CONTROL DE TOPOLOGÍA SOPORTADO POR TÉCNICAS DE CLUSTERING APLICADO A REDES AD HOC

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA
2007-08

TRABAJO DE INVESTIGACIÓN

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense
MADRID

Alumno:
Alberto Benito Peral

Director:
Luis Javier García Villalba

Control De Topología Soportado Por Técnicas De Clustering Aplicado A Redes Ad Hoc

Por Alberto Benito Peral

Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431
Universidad Complutense de Madrid (UCM)
C/ Profesor José García Santesmases s/n
Ciudad Universitaria, 28040 Madrid
E-mail: albertobp@fdi.ucm.es

Abstract

A MANET is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves, i.e., routing functionality will be incorporated into mobile nodes. Our routing protocol tries to give a satisfactory solution to these problems. It fulfils a hierarchical routing by means of the creation of clusters using AI algorithms (neural networks, fuzzy logic) adapted to a distributed stage. These skills allow to reduce the sending of control packets, as well as to find routes of more efficient form and to adapt better to the topological changes.

Keywords: MANET, CLUSTER, ROUTING, PROTOCOL, AD HOC.

Agradecimientos

Gracias al profesor Luis Javier García Villalba por guiarme en el desarrollo de este proyecto, así como al resto de personas que han colaborado en él. Gracias a nuestros familiares y amigos por apoyarnos durante este duro año. Gracias a esas personas que nos dan ese empujoncito que a veces todos necesitamos.

Índice

I. INTRODUCCIÓN

1.MOTIVACIÓN DEL PROYECTO

Dentro del espectacular desarrollo de las redes basadas en 802.11, la siguiente frontera que se está planteando son las redes autoorganizadas multisalto. Esta área se encuentra actualmente en plena efervescencia, tanto en estandarización como en cuanto a sistemas propietarios ofrecidos por los fabricantes. La finalidad del desarrollo de este tipo de redes es proporcionar aplicaciones y servicios ubicuos, siendo el Internet ubicuo su último fin.

El principal objetivo de las aplicaciones ubicuas es el establecimiento de un entorno donde los dispositivos con capacidad de procesamiento y comunicaciones (teléfonos móviles, PDA, dispositivos sensores, electrodomésticos, libros electrónicos, etc.) puedan comunicarse de forma inteligente y consciente con el entorno que les rodea de forma transparente para el usuario. Los sistemas de comunicaciones y sobre todo las redes inalámbricas ad hoc (Mobile ad hoc Networks - MANETs) se presentan como una tecnología de comunicación ideal para este tipo de entornos y aplicaciones. En este artículo presentamos una aplicación experimental como ejemplo de utilización de las tecnologías inalámbricas Bluetooth e IEEE 802.11 en el área de la computación ubicua.

El término computación ubicua, es un término propuesto por Mark Weiser, e indica el objetivo de hacer disponible y a la vez invisible al usuario el uso de sistemas de cálculo en el entorno en el que el usuario se encuentra. Los continuos avances tecnológicos han incentivado el desarrollo de dispositivos con capacidades de comunicación inalámbrica cada vez más pequeños, más potentes y con un consumo de batería más eficiente que hacen que cada día sea más realista el concepto de comunicación ubicua.

Fuertemente ligado al concepto de comunicación ubicua, encontramos las aplicaciones dependientes del entorno también conocidas como aplicaciones context-aware. Dichas aplicaciones se caracterizan por ser capaces de adaptar sus funciones de forma transparente en función del contexto, del tipo de usuario y del dispositivo utilizado.

En el área de las aplicaciones ubicuas las comunicaciones juegan un papel fundamental. En concreto, las características de las redes inalámbricas ad hoc pueden ofrecer una gran flexibilidad al sistema de comunicaciones. Las redes ad hoc, son redes inalámbricas que no requieren ningún tipo de infraestructura fija ni administración centralizada, donde las estaciones, además de ofrecer funcionalidades de estación final deben proporcionar también servicios de encaminamiento retransmitiendo paquetes entre aquellas estaciones que no tienen conexión inalámbrica directa.

Estas redes requieren nuevos algoritmos, protocolos y middleware, que superan las limitaciones anteriormente presentadas y permitan establecer redes independientes y descentralizadas. Dichos protocolos, deberían ser completamente

adaptativos, anticipando el comportamiento futuro de la red a partir de parámetros tales como el nivel de congestión, la tasa de errores, los cambios de rutas utilizadas, etc.

En los últimos años, especialmente en la última década, se han realizado grandes esfuerzos en el desarrollo de protocolos para MANETs. Sin embargo, esta cuestión no ha sido solucionada por completo. Entre los problemas surgidos hasta ahora, cabría destacar como los tres aspectos más decisivos la latencia de las comunicaciones, la sobrecarga ocasionada por paquetes de control y la escalabilidad. Los resultados proporcionados por cada propuesta dependen de los escenarios empleados para los estudios.

En esta tesis se presenta una nueva propuesta de protocolo para MANETs que pretende solucionar los problemas conocidos hasta el momento.

2. REDES DE TELECOMUNICACIONES EN EL SIGLO XXI

Anteriormente, hemos definido a las MANETs como redes ad hoc móviles. Esta definición, junto a otras características mencionadas o que mencionarmeos más aldeante, como autoorganización, Mobile IP, protocolos de enrutamiento, etc, puede dar lugar a un concepto bastante abstracto para quien no esté familiarizado con ellas. Es por ello que en esta sección vamos a dar una visión general del contexto en el que se sitúan las redes MANET. En este repaso, partiremos del concepto general de redes de computadores, para, poco a poco, ir estrechando los límites de nuestro ámbito, hasta llegar al concepto de Mobile IP. Todos los conceptos desarrollados a continuación serán mencionados frecuentemente a lo largo de la tesis.

2.1. Redes de computadores

Una red de computadores son dos o más computadores enlazados para el intercambio de datos. El software de una red permite compartir periféricos tales como Módem, Fax, CD-ROM, sistema de almacenamiento masivo, correo electrónico, manejo de proyectos en grupo, compartir aplicaciones, obtener recursos comunes, entre otros. La conexión física entre los computadores puede efectuarse por un alambre de cobre, fibra óptica, cableado utp, satélites de comunicación, microondas, entre otros.

- **Red pública**: una red pública se define como una red que puede usar cualquier persona y no como las redes que están configuradas con clave de acceso personal. Es una red de computadoras interconectadas, capaz de compartir información y que permite comunicar a usuarios sin importar su ubicación geográfica.

- **Red privada**: una red privada se definiría como una red que puede usarla solo algunas personas y que están configuradas con clave de acceso personal.

- **Red de área Personal (PAN)**: (Personal Area Network) es una red de ordenadores usada para la comunicación entre los dispositivos de la

computadora (teléfonos incluyendo las ayudantes digitales personales) cerca de una persona. Los dispositivos pueden o no pueden pertenecer a la persona en cuestión. El alcance de una PAN es típicamente algunos metros. Las PAN se pueden utilizar para la comunicación entre los dispositivos personales de ellos mismos (comunicación del intrapersonal), o para conectar con una red de alto nivel y el Internet (un up link). Las redes personales del área se pueden conectar con cables con los buses de la computadora tales como USB y FireWire. Una red personal sin hilos del área (WPAN) se puede también hacer posible con tecnologías de red tales como IrDA y Bluetooth.

- **Red de área local (LAN)**: una red que se limita a un área especial relativamente pequeña tal como un cuarto, un solo edificio, una nave, o un avión. Las redes de área local a veces se llaman una sola red de la localización. Nota: Para los propósitos administrativos, LANs grande se divide generalmente en segmentos lógicos más pequeños llamados los Workgroups. Un Workgroups es un grupo de las computadoras que comparten un sistema común de recursos dentro de un LAN.

- **Red del área del campus (CAN)**: Se deriva a una red que conecta dos o más LANs los cuales deben estar conectados en un área geográfica específica tal como un campus de universidad, un complejo industrial o una base militar.

- **Red de área metropolitana (MAN)**: una red que conecta las redes de un área dos o más locales juntos pero no extiende más allá de los límites de la ciudad inmediata, o del área metropolitana. Las rebajadoras múltiples, los interruptores y los cubos están conectados para crear a una MAN.

- **Red de área amplia (WAN)**: es una red de comunicaciones de datos que cubre un área geográfica relativamente amplia y que utiliza a menudo las instalaciones de transmisión proporcionadas por los portadores comunes, tales como compañías del teléfono. Las tecnologías WAN funcionan generalmente en las tres capas más bajas del Modelo de referencia OSI: la capa física, la capa de transmisión de datos, y la capa de red.

2.1.1. Internet

Algunos definen Internet como "La Red de Redes", y otros como "Las Autopistas de la Información".

Efectivamente, Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores, o sea de unos pocos ordenadores en un mismo edificio o empresa. Además, ésta es "La Red de Redes" porque es la más grande. Prácticamente todos los países del mundo tienen acceso a Internet. En algunos, como los del Tercer Mundo, sólo acceden los multimillonarios y en otros como USA o los países más desarrollados de Europa, no es difícil conectarse.

Por la Red Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información. Hay 50 millones de "Internautas", es decir, de personas que "navegan" por Internet en todo el

Mundo. Se dice "navegar" porque es normal el ver información que proviene de muchas partes distintas del Mundo en una sola sesión.

Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras. Incluso podemos ver conectados a la Red cámaras de vídeo, robots, y máquinas de refrescos.

2.1.1.1. Historia de Internet

Internet nació en EE.UU. hace unos 30 años. Un proyecto militar llamado ARPANET pretendía poner en contacto una importante cantidad de ordenadores de las instalaciones del ejército de EE.UU. Este proyecto gastó mucho dinero y recursos en construir la red de ordenadores más grande en aquella época.

Al cabo del tiempo, a esta red se fueron añadiendo otras empresas. Así se logró que creciera por todo el territorio de EE.UU. Hará unos 10 años se conectaron las instituciones públicas como las Universidades y también algunas personas desde sus casas. Fue entonces cuando se empezó a extender Internet por los demás países del Mundo, abriendo un canal de comunicaciones entre Europa y EE.UU.

Internet crece a un ritmo vertiginoso. Constantemente se mejoran los canales de comunicación con el fin de aumentar la rapidez de envío y recepción de datos. Cada día que pasa se publican en la Red miles de documentos nuevos, y se conectan por primera vez miles de personas. Con relativa frecuencia aparecen nuevas posibilidades de uso de Internet, y constantemente se están inventando nuevos términos para poder entenderse en este nuevo mundo que no para de crecer.

2.1.1.2. Servicios de Internet

Las posibilidades que ofrece Internet se denominan servicios. Cada servicio es una manera de sacarle provecho a la Red independiente de las demás. Una persona podría especializarse en el manejo de sólo uno de estos servicios sin necesidad de saber nada de los otros. Sin embargo, es conveniente conocer todo lo que puede ofrecer Internet, para poder trabajar con lo que más nos interese.

Hoy en día, los servicios más usados en Internet son: Correo Electrónico, World Wide Web, FTP, Grupos de Noticias, IRC y Servicios de Telefonía.

El Correo Electrónico nos permite enviar cartas escritas con el ordenador a otras personas que tengan acceso a la Red. Las cartas quedan acumuladas en Internet hasta el momento en que se piden. Es entonces cuando son enviadas al ordenador del destinatario para que pueda leerlas. El correo electrónico es casi instantáneo, a diferencia del correo normal, y además muy barato. Podemos cartearnos con cualquier persona del Mundo que disponga de conexión a Internet.

La World Wide Web, o WWW como se suele abreviar, se inventó a finales de los 80 en el CERN, el Laboratorio de Física de Partículas más importante del Mundo. Se trata de un sistema de distribución de información tipo revista. En la Red quedan

almacenadas lo que se llaman Páginas Web, que no son más que páginas de texto con gráficos o fotos. Aquellos que se conecten a Internet pueden pedir acceder a dichas páginas y acto seguido éstas aparecen en la pantalla de su ordenador. Este sistema de visualización de la información revolucionó el desarrollo de Internet. A partir de la invención de la WWW, muchas personas empezaron a conectarse a la Red desde sus domicilios, como mero entretenimiento. Internet recibió un gran impulso, hasta el punto de que hoy en día casi siempre que se hablamos de Internet, nos referimos a la WWW.

El FTP (File Transfer Protocol) nos permite enviar ficheros de datos por Internet. Ya no es necesario guardar la información en disquetes para usarla en otro ordenador. Con este servicio, muchas empresas informáticas han podido enviar sus productos a personas de todo el mundo sin necesidad de gastar dinero en miles de disquetes ni envíos. Muchos particulares hacen uso de este servicio para por ejemplo dar a conocer sus creaciones informáticas a nivel mundial.

Los Grupos de Noticias son el servicio más apropiado para entablar debate sobre temas técnicos. Se basa en el servicio de Correo Electrónico. Los mensajes que enviamos a los Grupos de Noticias se hacen públicos y cualquier persona puede enviarnos una contestación. Este servicio es de gran utilidad para resolver dudas difíciles, cuya respuesta sólo la sepan unas pocas personas en el mundo.

El servicio IRC (Internet Relay Chat) nos permite entablar una conversación en tiempo real con una o varias personas por medio de texto. Todo lo que escribimos en el teclado aparece en las pantallas de los que participan de la charla. También permite el envío de imágenes u otro tipo de ficheros mientras se dialoga.

Los Servicios de Telefonía son las últimas aplicaciones que han aparecido para Internet. Nos permiten establecer una conexión con voz entre dos personas conectadas a Internet desde cualquier parte del mundo sin tener que pagar el coste de una llamada internacional. Algunos de estos servicios incorporan no sólo voz, sino también imagen. A esto se le llama Videoconferencia.

Internet dispone de otros servicios menos usados, por haberse quedado anticuados, o bien por tener sólo aplicaciones muy técnicas. Algunos de estos son: Archie, Gopher, X.500, WAIS y Telnet.

El servicio Archie es un complemento del FTP. Sirve para buscar ficheros concretos por la Red, para más tarde cogerlos por FTP.

Gopher es el antecesor de la WWW. Es un sistema de obtención de información que usa la técnica de la navegación, como la WWW, pero carece de los elementos multimedia, esto es imágenes y sonido principalmente, que da tanto impulso a la WWW. Este servicio aún esta disponible en Internet, sin embargo no hay mucha gente que lo use.

X.500 y WAIS son servicios de búsqueda de personas y datos sobre esas personas. Este servicio se usa en Instituciones públicas como Universidades para la localización de Investigadores y para averiguar en que proyectos están trabajando.

Con Telnet podemos tomar el control de un ordenador conectado a la Red, de manera remota, o sea, a distancia. Es de gran utilidad para trabajar con grandes ordenadores en empresas o instituciones, en las que muchos usuarios acceden al mismo tiempo a un ordenador central de gran potencia.

2.1.2. Redes inalámbricas

Desde hace relativamente poco tiempo, se está viviendo lo que puede significar un revolución en el uso de las tecnologías de la información tal y como lo conocemos. Esta revolución puede llegar a tener una importancia similar a la que tuvo la adopción de Internet por el gran público.

De una forma discreta, las redes inalámbricas o Wireless Networks (WN), se están introduciendo en el mercado de consumo gracias a unos precios populares y a un conjunto de entusiastas, mayoritariamente particulares, que han visto las enormes posibilidades de esta tecnología.

Las aplicaciones de las redes inalámbricas son infinitas. De momento van a crear una nueva forma de usar la información, pues ésta estará al alcance de todos a través de Internet en cualquier lugar (en el que haya cobertura).

En un futuro cercano se reunificarán todo aquellos dispositivos con los que hoy contamos para dar paso a unos nuevos que perfectamente podrían llamarse Terminales Internet en los cuales estarían reunidas las funciones de teléfono móvil, agenda, terminal de vídeo, reproductor multimedia, ordenador portátil y un largo etcétera.

Se podría dar lugar a una Internet paralela y gratuita la cual estaría basada en las redes que altruistamente cada uno de nosotros pondríamos a disposición de los demás al incorporarnos a las mismas como destino y origen de la información.

En un futuro también cercano la conjugación de las redes Mesh, con las redes inalámbricas y las redes Grid podría llevar a cabo al nacimiento de nuevas formas de computación que permitan realizar cálculos inimaginables hasta ahora debido a las necesidades HW de las que eran objeto.

En las grandes ciudades por fin se podría llevar a cabo un control definitivo del tráfico con el fin de evitar atascos, limitando la velocidad máxima y/o indicando rutas alternativas en tiempo real.

Las tecnologías que son necesarias para llevar a cabo estos sistemas hoy existen desde ayer, su precio es mínimo o al menos muy asequible y su existencia mañana sólo depende de las estrategias comerciales de las empresas que las poseen.

Antes de echar la imaginación a volar es necesario tener un cierto conocimiento sobre la tecnología que va a ser la base de estas aplicaciones, sobre las redes inalámbricas. Vamos entonces a intentar describir las mismas.

Lo primero que tenemos que hacer antes que nada es situarnos dentro del mundo inalámbrico. Para ello vamos a hacer una primera clasificación que nos centre ante las diferentes variantes que podemos encontrarnos:

- Redes inalámbricas personales
- Redes inalámbricas 802.11
- Redes inalámbricas de consumo

2.1.2.1. Redes inalámbricas personales

Dentro del ámbito de estas redes podemos integrar a dos principales actores:

- En primer lugar y ya conocido por bastantes usuarios están las redes que se usan actualmente mediante el intercambio de información mediante infrarrojos. Estas redes son muy limitadas dado su corto alcance, necesidad de "visión sin obstáculos" entre los dispositivos que se comunican y su baja velocidad (hasta 115 kbps). Se encuentran principalmente en ordenadores portátiles, PDAs (Agendas electrónicas personales), teléfonos móviles y algunas impresoras.
- En segundo lugar el Bluetooth, estándar de comunicación entre pequeños dispositivos de uso personal, como pueden ser los PDAs, teléfonos móviles de nueva generación y algún que otro ordenador portátil. Su principal desventaja es que su puesta en marcha se ha ido retrasando desde hace años y la aparición del mismo ha ido plagada de diferencias e incompatibilidades entre los dispositivos de comunicación de los distintos fabricantes que ha imposibilitado su rápida adopción. Opera dentro de la banda de los 2'4 Ghz.

2.1.2.2. Redes inalámbricas de consumo

Existen dos tecnologías fundamentales de redes inalámbricas en el mundo comercial

- **Redes CDMA** (estándar de telefonía móvil estadounidense) y GSM (estándar de telefonía móvil europeo y asiático). Son los estándares que usa la telefonía móvil empleados alrededor de todo el mundo en sus diferentes variantes.

- **802.16** son redes que pretenden complementar a las anteriores estableciendo redes inalámbricas metropolitanas (MAN) en la banda de entre los 2 y los 11 Ghz.

Las redes inalámbricas o WN básicamente se diferencian de las redes conocidas hasta ahora por el enfoque que toman de los niveles más bajos de la pila OSI, el nivel físico y el nivel de enlace, los cuales se definen por el 802.11 del IEEE (Organismo de estandarización internacional).

Como suele pasar siempre que un estándar aparece y los grandes fabricantes se interesan por él, aparecen diferentes aproximaciones al mismo lo que genera una incipiente confusión.

Nos encontramos ante tres principales variantes:

- **802.11a**: fue la primera aproximación a las **WN** y llega a alcanzar velocidades de hasta 54 Mbps dentro de los estándares del IEEE y hasta 72 y 108 Mbps con tecnologías de desdoblamiento de la velocidad ofrecidas por diferentes

fabricantes, pero que no están (a día de hoy) estandarizadas por el IEEE. Esta variante opera dentro del rango de los 5 Ghz. Inicialmente se soportan hasta 64 usuarios por Punto de Acceso. Sus principales ventajas son su velocidad, la base instalada de dispositivos de este tipo, la gratuidad de la frecuencia que usa y la ausencia de interferencias en la misma. Sus principales desventajas son su incompatibilidad con los estándares 802.11b y g, la no incorporación a la misma de QoS, la no disponibilidad de esta frecuencia en Europa dado que esta frecuencia está reservada a la HyperLAN2 y la parcial disponibilidad de la misma en Japón. El hecho de no estar disponible en Europa prácticamente la descarta de nuestras posibilidades de elección para instalaciones en este continente.

- **802.11b**: es la segunda aproximación de las WN. Alcanza una velocidad de 11 Mbps estandarizada por el IEEE y una velocidad de 22 Mbps por el desdoblamiento de la velocidad que ofrecen algunos fabricantes pero sin la estandarización (a día de hoy) del IEEE. Opera dentro de la frecuencia de los 2'4 Ghz. Inicialmente se soportan hasta 32 usuarios por PA. Adolece de varios de los inconvenientes que tiene el 802.11a como son la falta de QoS, además de otros problemas como la masificación de la frecuencia en la que transmite y recibe, pues en los 2'4 Ghz funcionan teléfonos inalámbricos, teclados y ratones inalámbricos, hornos microondas, dispositivos Bluetooth..., lo cual puede provocar interferencias. En el lado positivo está su rápida adopción por parte de una gran comunidad de usuarios debido principalmente a unos muy bajos precios de sus dispositivos, la gratuidad de la banda que usa y su disponibilidad gratuita alrededor de todo el mundo. Está estandarizado por el IEEE.
- **802.11g**: Es la tercera aproximación a las WN, y se basa en la compatibilidad con los dispositivos 802.11b y en el ofrecer unas velocidades de hasta 54 Mbps. A 05/03/2003 se encuentra en estado de borrador en el IEEE, se prevee que se estandarice para mediados de 2003. Funciona dentro de la frecuencia de 2'4 Ghz. Dispone de los mismos inconvenientes que el 802.11b además de los que pueden aparecer por la aún no estandarización del mismo por parte del IEEE. Las ventajas de las que dispone son las mismas que las del 802.11b además de su mayor velocidad.

Sea cual sea el estándar que elijamos vamos a disponer principalmente de dos tipos de dispositivos:

- **Dispositivos "Tarjetas de red", o TR**, que serán los que tengamos integrados en nuestro ordenador, o bien conectados mediante un conector PCMCIA ó USB si estamos en un portátil o en un slot PCI si estamos en un ordenador de sobremesa. Ssubstituyen a las tarjetas de red Ethernet o Token Ring a las que estábamos acostumbrados. Recibirán y enviarán la información hacia su destino desde el ordenador en el que estemos trabajando. La velocidad de transmisión / recepción de los mismos es variable dependiendo del fabricante y de los estándares que cumpla.
- **Dispositivos "Puntos de Acceso", ó PA**, los cuales serán los encargados de recibir la información de los diferentes TR de los que conste la red bien para su centralización bien para su encaminamiento. Complementan a los Hubs, Switches o Routers, si bien los PAs pueden substituir a los últimos pues muchos de ellos ya incorporan su funcionalidad. La velocidad de transmisión / recepción

de los mismos es variable, las diferentes velocidades que alcanzan varían según el fabricante y los estándares que cumpla.

Para una representación gráfica de una red inalámbrica vea el siguiente gráfico.

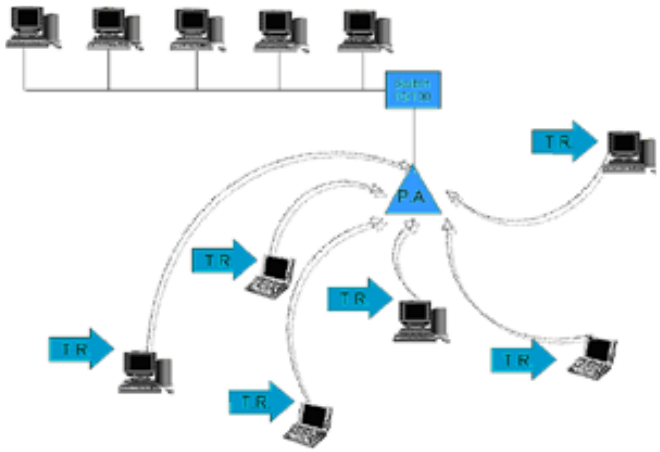


Figura 2.1. *Red inalámbrica.*

2.1.3. WiMAX

WiMAX son las siglas de 'Worldwide Interoperability for Microwave Access', y es la marca que certifica que un producto está conforme con los estándares de acceso inalámbrico 'IEEE 802.16′. Estos estándares permitirán conexiones de velocidades similares al ADSL o al cablemódem, sin cables, y hasta una distancia de 50-60 km. Este nuevo estándar será compatible con otros anteriores, como el de Wi-Fi (IEEE 802.11).

WiMAX puede proveer de acceso de banda ancha Wíreless de hasta 50 Kilómetros. Si lo comparamos con el protocolo Wíreless 802.11, el cual está limitado en la mayoría de las ocasiones a unos 100 Metros, nos damos cuenta de la gran diferencia que separa estas dos tecnologías inalámbricas. De hecho se suele llamar a WiMAX como "Wifi con esteroides".

Algunas de las ventajas de WiMAX son:

- Puede dar cobertura a un área bastante extenso y la instalación de las antenas para transmitir y recibir, formando estaciones base, son sencillas y rápidas de instalar. Esto lo hace adecuado para dar comunicación en ciudades enteras, pudiendo formar una MAN, en lugar de un área de red local como puede proporcionar Wifi.
- WiMAX tiene una velocidad de transmisión mayor que la de Wifi, y dependiendo del ancho de banda disponible, puede producir transmisiones de hasta 70 MB comparado con los 54 MB que puede proporcionar Wifi.

- Puede ser simétrico lo cual significa que puede proporcionar un flujo de datos similar tanto de subida como de bajada.
- Las antenas de WiMAX operan a una frecuencia de hasta 60 mHz. Un detalle a tener en cuenta es que las antenas no tienen que estar directamente alineadas con sus clientes.

WiMAX está pensado para construir una infraestructura de red cuando el entorno o distancia no es favorable para una red cableada. Es una alternativa más rápida y barata que tener que instalar cables.

Cabe destacar el llamado WiMAX forum el cual es un grupo de empresas que se encargan de diseñar las normas y estándares de la tecnología WiMAX y ha probar todos los nuevos componentes que van surgiendo. Actualmente lo forman más de 100 compañías y seguirá aumentando.

Dentro de WiMAX debemos hacer una pequeña diferenciación. El estándar 802.16d para terminales fijos, y el 802.16e para estaciones en movimiento. Esto marca una distinción en la manera de usar este protocolo, aunque lo ideal es utilizar una combinación de ambos.

2.2. Protocolos de comunicaciones

Los protocolos de comunicaciones definen las reglas para la transmisión y recepción de la información entre los nodos de la red, de modo que para que dos nodos se puedan comunicar entre si es necesario que ambos empleen la misma configuración de protocolos. Entre los protocolos propios de una red de área local podemos distinguir dos principales grupos. Por un lado están los protocolos de los niveles físico y de enlace, niveles 1 y 2 del modelo OSI, que definen las funciones asociadas con el uso del medio de transmisión: envío de los datos a nivel de bits y trama, y el modo de acceso de los nodos al medio. Estos protocolos vienen unívocamente determinados por el tipo de red (Ethernet, Token Ring, etc.). El segundo grupo de protocolos se refiere a aquellos que realizan las funciones de los niveles de red y transporte, niveles 3 y 4 de OSI, es decir los que se encargan básicamente del encaminamiento de la información y garantizar una comunicación extremo a extremo libre de errores. Estos protocolos transmiten la información a través de la red en pequeños segmentos llamados paquetes. Si un ordenador quiere transmitir un fichero grande a otro, el fichero es dividido en paquetes en el origen y vueltos a ensamblar en el ordenador destino. Cada protocolo define su propio formato de los paquetes en el que se especifica el origen, destino, longitud y tipo del paquete, así como la información redundante para el control de errores. Los protocolos de los niveles 1 y 2 dependen del tipo de red, mientras que para los niveles 3 y 4 hay diferentes alternativas, siendo TCP/IP la configuración mas extendida. Lo que la convierte en un estándar de facto. Por su parte, los protocolos OSI representan una solución técnica muy potente y flexible, pero que actualmente esta escasamente implantada en entornos de red de área local. La jerarquía de protocolo OSI.

2.2.1. Paquetes de información

La información es embalada en sobres de datos para la transferencia. Cada grupo, a menudo llamados paquetes incluyen las siguientes informaciones:

- **Datos a la carga**: La información que se quiere transferir a través de la red, antes de ser añadida ninguna otra información. El termino carga evoca a la pirotecnia, siendo la pirotecnia una analogía apropiada para describir como los datos son disparados de un lugar a otro de la red.

- **Dirección**: El destino del paquete. Cada segmento de la red tiene una dirección, que solamente es importante en una red que consista en varias LAN conectadas. También hay una dirección de la estación y otra de la aplicación. La dirección de la aplicación se requiere para identificar a que aplicación de cada estación pertenece el paquete de datos.

- **Código de control**: Informa que describe el tipo de paquete y el tamaño. Los códigos de control también códigos de verificación de errores y otra información

2.3. Pila de protocolos TCP/IP

Cada nivel de la jerarquía de protocolos OSI tiene una función especifica y define un nivel de comunicaciones entre sistemas. Cuando se define un proceso de red, como la petición de un archivo por un servidor, se empieza en el punto desde el que el servidor hizo la petición. Entonces, la petición va bajando a través de la jerarquía y es convertida en cada nivel para poder ser enviada por la red.

- **Nivel Físico:** Define las características físicas del sistema de cableado, abarca también los métodos de red disponibles, incluyendo Token Ring, Ethernet y ArcNet. Este nivel especifica lo siguiente:
    - Conexiones eléctricas y físicas.
    - Como se convierte en un flujo de bits la información que ha sido paquetizada.
    - Como consigue el acceso al cable la tarjeta de red.
- **Nivel de Enlace de Datos:** Define las reglas para enviar y recibir información a través de la conexión física entre dos sistemas.
- **Nivel de Red:** Define protocolos para abrir y mantener un camino entre equipos de la red. Se ocupa del modo en que se mueven los paquetes.
- **Nivel de Transporte:** Suministra el mayor nivel de control en el proceso que mueve actualmente datos de un equipo a otro.
- **Nivel de Sesión:** Coordina el intercambio de información entre equipos, se llama así por la sesión de comunicación que establece y concluye.
- **Nivel de Presentación:** En este los protocolos son parte del sistema operativo y de la aplicación que el usuario acciona en la red.
- **Nivel de Aplicación:** En este el sistema operativo de red y sus aplicaciones se hacen disponibles a los usuarios. Los usuarios emiten ordenes para requerir los servicios de la red.

2.4. Internet Protocol (IP)

Desde principios de los 90, la IETF (Internet Engineering Task Force) investiga opciones para reemplazar la clásica versión 4 del protocolo IP, a fin de

subsanar los problemas que se van detectando: falta de suficientes direcciones IP, excesivo volumen de las tablas de encaminamiento, poca atención al tipo de tráfico cursado, escasa seguridad y otros varios. En 1995 se presenta la versión 6 de IP y desde entonces sus características nuevas siguen sometidas a debate y discusión. Entre estas novedades destacan: nuevo formato de direccionamiento, aceleración del encaminamiento, soporte a la movilidad de máquinas, soporte al tráfico de tiempo real y seguridad integrada en el protocolo.

De esta forma nace la nueva versión del protocolo IP, llamada IPv6, aunque también es conocida como IPng (Internet Protocol Next Generation). Es la versión 6, debido a que la numero 5 no pasó de la fase experimental. La compatibilidad con la versión 4 es prácticamente total, ya que se han incluido características de compatibilidad. Algunas de las modificaciones, están encaminadas a mejorar la seguridad en la red, que apenas existia en la versión 4. En este trabajo se realizará una pequeña mención a la versión 4 de IP, que permitirá establecer un marco de comparación entre ambas versiones.

2.4.1. IPv4

Es importante aclarar que de acuerdo al modelo OSI, es la CAPA DE RED la que se encarga de controlar la comunicación entre un equipo y otro. Esta conforma los paquetes IP que serán enviados por la capa inferior. Desencapsula los paquetes recibidos pasando a la capa superior la información dirigida a una aplicación.

El Protocolo IP proporciona un sistema de distribución que es poco fiable incluso en una base solida. El protocolo IP especifica que la unidad básica de transferencia de datos en el TCP/IP es el datagrama.

Los datagramas pueden ser retrasados, perdidos, duplicados, enviados o bien en una secuencia incorrecta, o bien fragmentados intencionadamente para permitir que un nodo con un buffer limitado pueda coger todo el datagrama. Es la responsabilidad del protocolo IP reensamblar los fragmentos del datagrama en el orden correcto. En algunas situaciones de error los datagramas son descartados sin mostrar ningún mensaje mientras que en otras situaciones los mensajes de error son recibidos por la maquina origen (esto lo hace el protocolo ICMP). El protocolo ICMP (Internet Control Message Protocol), proporciona el medio para que el software de hosts y gateways intermedios se comuniquen. El protocolo ICMP tiene su propio numero de protocolo (numero 1), que lo habilita para utilizar el IP directamente. La implementacion de ICMP es obligatoria como un subconjunto lógico del protocolo IP. Los mensajes de error de este protocolo los genera y procesa TCP/IP, y no el usuario.

El protocolo IP también define cual sera la ruta inicial por la que serán mandados los datos.

Cuando los datagramas viajan de unos equipos a otros, es posible que atraviesen diferentes tipos de redes. El tamaño máximo de estos paquetes de datos puede variar de una red a otra, dependiendo del medio físico que se emplee para su transmisión. A este tamaño máximo se le denomina MTU (Maximum Transmission Unit), y ninguna red puede transmitir un paquete de tamaño mayor a esta MTU. El datagrama consiste en una cabecera y datos.

- **Longitud de la Cabecera.** Este campo ocupa 4 bits, y representa el número de octetos de la cabecera dividido por cuatro, lo que hace que este sea el numero de grupos de 4 octetos en la cabecera.
- **Versión.** El campo versión ocupa 4 bits. Este campo hace que diferentes versiones del protocolo IP puedan operar en la Internet. En este caso se trata de la versión 4.
- **Tipo de servicio.** Este campo ocupa un octeto de la cabecera IP, y especifica la precedencia y la prioridad del datagrama IP. Los tres primeros bits del octeto indican la precedencia. Los valores de la precedencia pueden ser de 0 a 7. Cero es la precedencia normal, y 7 esta reservado para control de red. Muchos Gateways ignoran este campo. Los otros 4 bits definen el campo prioridad, que tiene un rango de 0 a 15. Las cuatro prioridades que están asignadas son: 0, (por defecto, servicio normal), 1 (minimizar el costo monetario), 2 (máxima fiabilidad), 4 (Maximizar la transferencia), 8 (El bit +4 igual a 1, define minimizar el retraso). Estos valores son utilizados por los routers para direccionar las solicitudes de los usuarios.
- **Longitud Total.** Este campo se utiliza para identificar el número de octetos en el datagrama total.
- **Identificación.** El valor del campo identificación es un numero secuencial asignado por el Host origen. El campo ocupa dos octetos. Los números oscilan entre 0 y 65.535, que cuando se combinan con la dirección del Host forman un número único en la Internet. El número se usa para ayudar en el reensamblaje de los fragmentos de datagramas.
- **Fragmentos Offset.** Cuando el tamaño de un datagrama excede el MTU, este se segmenta. El fragmento Offset representa el desplazamiento de este segmento desde en inicio del datagrama entero.
- **Flags.** El campo flag ocupa 3 bits y contiene dos flags. El bit +5 del campo flags se utiliza para indicar el último datagrama fragmentado cuando toma valor cero. El bit +7 lo utiliza el servidor origen para evitar la fragmentación. Cuando este bit toma valor diferente de cero y la longitud de un datagrama excede el MTU, el datagrama es descartado y un mensaje de error es enviado al Host de origen por medio del protocolo ICMP.
- **Tiempo de Vida.** El campo tiempo de vida ocupa un octeto. Representa el número máximo de segundos que un datagrama puede existir en Internet, antes de ser descartado. Un Datagrama puede existir un máximo de 255 segundos. El número recomendado para IP es 64. El originador del datagrama manda un mensaje ICMP cuando el datagrama es descartado.
- **Protocolo.** El campo protocolo se utiliza para identificar la capa de mayor nivel más cercana usando el IP. Este es un campo de 0 bits, que normalmente identifica tanto la capa TCP (valor 6), como la capa UDP (valor 17) en el nivel de transporte, pero puede identificar hasta 255 protocolos de la capa de transporte.
- **Checksum.** El checksum proporciona la seguridad de que el datagrama no ha sido dañado ni modificado. Este campo tiene una longitud de 16 bits. El checksum incluye todos los campos de todos los campos de la cabecera IP, incluido el mismo, cuyo valor es cero a efectos de cálculo.

Un Gateway o nodo que efectúe alguna modificación en los campos de la cabecera (por ejemplo en el tiempo de vida), debe recalcular el valor del checksum antes de enviar el datagrama.

Los usuarios del IP deben proporcionar su propia integridad en los datos, ya que el checksum es solo para la cabecera.

- Dirección de Origen. Este campo contiene un identificador de red (Netid) y un identificador de Host (Hostid). El campo tiene una longitud de 32 bits. La dirección puede ser de clase A, B, C. (ver Direcciones IP).
- Dirección de Destino. Este campo contiene el Netid y el Hostid del destino. El campo tiene una longitud de 32 bits. La dirección puede ser de clase A, B, C o D (ver Direcciones IP).
- Opciones. La existencia de este campo viene determinada por la longitud de la cabecera. Si esta es mayor de cinco, por lo menos existe una opción. Aunque un Host no esta obligado a poner opciones, puede aceptar y procesar opciones recibidas en un datagrama. El campo Opciones es de longitud variable. Cada octeto esta formado por los campos Copia, Clase de Opción y Número de Opción.
- Copia. El campo Copia sirve para que cuando un datagrama va a ser fragmentado y viaja a través de nodos o Gateways. Cuando tiene valor 1, las opciones son las mismas para todos los fragmentos, pero si toma valor 0, las opciones son eliminadas. Clase de Opción es un campo que cuando tiene valor 0, indica datagrama o control de red; Cuando tiene valor 2, indica depuración o medida. Los valores 1 y 3 están reservados para un uso futuro. El Número de Opción indica una acción específica.
- Padding. Cuando esta presente el campo Pad, consiste en 1 a 3 octetos puestos a cero, si es necesario, para hacer que el número total de octetos en la cabecera sea divisible por cuatro.
- Datos. El campo datos consiste en una cadena de octetos. Cada octeto tiene un valor entre 0 y 255. El tamaño de la cadena puede tener un mínimo y un máximo, dependiendo del medio físico. El tamaño máximo esta definido por la longitud total del datagrama. El tamaño del campo Datos en octetos es igual a:

(Longitud Total del Datagrama) - (Longitud de la cabecera)

Direcciones IP y máscaras de red

En una red TCP/IP los ordenadores se identifican mediante un número que se denomina dirección IP. Esta dirección ha de estar dentro del rango de direcciones asignadas al organismo o empresa a la que pertenece, estos rangos son concedidos por un organismo central de Internet, el NIC (Network Information Center).

Las direcciones IP hacen que el envío de datos entre computadores se haga de forma eficaz, de un modo similar al que se utilizan los números de teléfonos. Una dirección IP está formada por 32 bits, que se agrupan en octetos.

2.4.2. IPv6

IPv6 es la versión 6 del Protocolo de Internet (IP por sus siglas en inglés, Internet Protocol), es el encargado de dirigir y encaminar los paquetes en la red, fue diseñado en los años 70 con el objetivo de interconectar redes.

El IPv6 fue diseñado por Steve Deering y Craig Mudge, adoptado por Internet Engineering Task Force (IETF) en 1994. IPv6 también se conoce por "IP Next Generation" o "IPng".

Esta nueva versión del Protocolo de Internet está destinada a sustituir al estándar IPv4, la misma cuenta con un límite de direcciones de red, lo cual impide el crecimiento de la red.

Portal go6

IPv6 empieza a ganar terreno en el mercado del gobierno federal de los E.E.U.U. y los portadores asiáticos de comunicaciones. El gobierno federal piensa incluir soporte IPv6 para sus redes antes del 2008.

El nuevo portal go6 incluye información más comprensiva sobre IPv6 en la web. Fue creado por Hexago, un vendedor canadiense de IPv6. Cuenta con experiencia en la implementación y aplicación de IPv6. Nos proveen acceso a las últimas herramientas e informaciones sobre la nueva versión del Protocolo de Internet.

A pesar de que IPv6 fue diseñado para ofrecer una seguridad mejor que Ipv4, la seguridad sigue siendo una edición en nuevas instalaciones debido a la escasez de las herramientas de seguridad para estos protocolos. Para ello podemos hacer uso de los cortafuegos (firewall) actuales.

La IPv4 vrs. IPv6

Actualmente se utiliza con más frecuencia la versión 4 del Protocolo de Internet, el aumento de usuarios, aplicaciones, servicios y dispositivos está provocando la migración a una nueva versión.

IPv4 soporta 4.294.967.296 (232) direcciones de red, este es un número pequeño cuando se necesita otorgar a cada computadora, teléfonos, PDA, autos, etc. IPv6 soporta 340.282.366.920.938.463.463.374.607.
431.768.211.456 (2128 ó 340 sextillones) direcciones de red.

Por lo general las direcciones IPv6 están compuestas por dos partes lógicas: un prefijo de 64 bits y otra parte de 64 bits que corresponde al identificador de interfaz, que casi siempre se genera automáticamente a partir de la dirección MAC (Media Access Control address) de la interfaz a la que está asignada la dirección.

Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP.

No debemos confundir la dirección MAC que es un número hexadecimal fijo, que es asignado a la tarjeta o dispositivo de red por el fabricante por la dirección IP, mientras que la dirección IP se puede cambiar.

Solución actual

La utilización de IPv6 se ha frenado por la Traducción de Direcciones de Red (NAT, Network Address Translation), temporalmente alivia la falta de estas direcciones de red.

Este mecanismo consiste en usar una dirección IPv4 para que una red completa pueda acceder a internet. Pero esta solución nos impide la utilización de varias aplicaciones, ya que sus protocolos no son capaces de atravesar los dispositivos NAT, por ejemplo P2P, voz sobre IP (VoIP), juegos multiusuarios, entre otros.

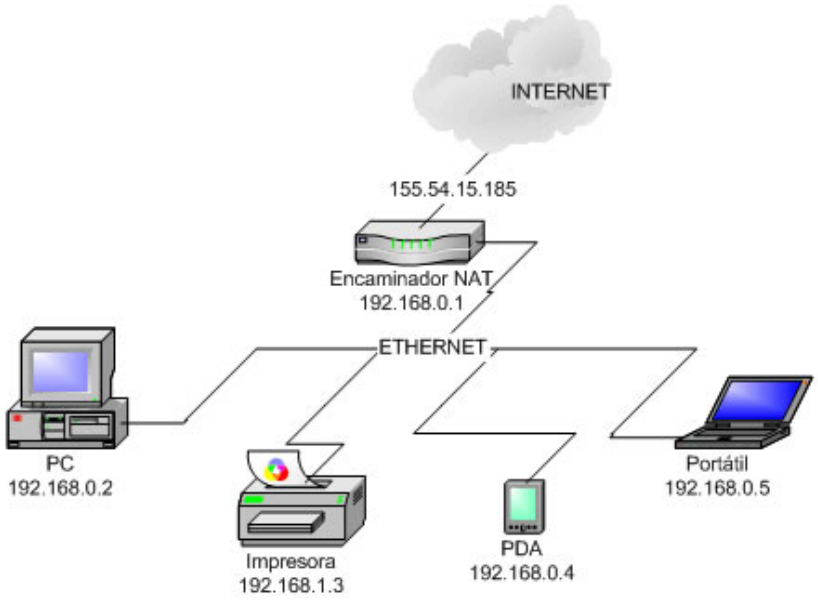


Figura 2.2. *Traducción de direcciones IPv6*

Características de la IPv6

Quizás las principales características de la IPv6 se síntetizan en el mayor espacio de direccionamiento, seguridad, autoconfiguración y movilidad. Pero también hay otras que son importantes mencionar:

- Infraestructura de direcciones y enrutamiento eficaz y jerárquica.
- Mejora de compatiblidad para Calidad de Servicio (QoS) y Clase de Servicio (CoS).
- Multicast: envío de un mismo paquete a un grupo de receptores.
- Anycast: envío de un paquete a un receptor dentro de un grupo.
- Movilidad: una de las características obligatorias de IPv6 es la posibilidad de conexión y desconexión de nuestro ordenador de redes IPv6 y, por tanto, el poder viajar con él sin necesitar otra aplicación que nos permita que ese enchufe/desenchufe se pueda hacer directamente.

- Seguridad Integrada (IPsec): IPv6 incluye IPsec, que permite autenticación y encriptación del propio protocolo base, de forma que todas las aplicaciones se pueden beneficiar de ello.
- Capacidad de ampliación.
- Calidad del servicio.
- Velocidad.

Tipos de direcciones IP

Unicast:

Este tipo de direcciones son bastante conocidas. Un paquete que se envía a una dirección unicast debería llegar a la interfaz identificada por dicha dirección.

Multicast:

Las direcciones multicast identifican un grupo de interfaces. Un paquete destinado a una dirección multicast llega a todos los los interfaces que se encuentran agrupados bajo dicha dirección.

Anycast:

Las direcciones anycast son sintácticamente indistinguibles de las direcciones unicast pero sirven para identificar a un conjunto de interfaces. Un paquete destinado a una dirección anycast llega a la interfaz "más cercana" (en términos de métrica de "routers"). Las direcciones anycast sólo se pueden utilizar en "routers".

Direcciones IPv6

La función de la dirección IPv6 es exactamente la misma a su predecesor IPv4, pero dentro del protocolo IPv6.

Está compuesta por 8 segmentos de 2 bytes cada uno, que suman un total de 128 bits, el equivalente a unos 3.4×1038 hosts direccionables. La ventaja con respecto a la dirección IPv4 es obvia en cuanto a su capacidad de direccionamiento.

Su representación suele ser hexadecimal y para la separación de cada par de octetos se emplea el símbolo ":". Un bloque abarca desde 0000 hasta FFFF. Algunas reglas acerca de la representación de direcciones IPv6 son:

- Los ceros iniciales, como en IPv4, se pueden obviar. Ejemplo: 2001:0123:0004:00ab:0cde:3403:0001:0063 -> 2001:123:4:ab:cde:3403:1:63.
- Los bloques contiguos de ceros se pueden comprimir empleando "::". Esta operación sólo se puede hacer una vez.
- Ejemplo: 2001:0:0:0:0:0:0:4 -> 2001::4.
- Ejemplo no válido: 2001:0:0:0:2:0:0:1 -> 2001::2::1 (debería ser 2001::2:0:0:1 ó 2001:0:0:0:2::1).

Paquetes IPv6

La cabecera se encuentra en los primeros 40 bytes del paquete, contiene las direcciones de origen y destino con 128 bits cada una, la versión 4 bits, la clase de tráfico 8 bits, etiqueta de flujo 20 bits, longitud del campo de datos 16 bits, cabecera siguiente 8 bits, y límite de saltos 8 bits.

¿Qué es un túnel IPv6 en IPv4?

Es un mecanismo de transición que permite a máquinas con IPv6 instalado comunicarse entre si a través de una red IPv4.

El mecanismo consiste en crear los paquetes IPv6 de forma normal e introducirlos en un paquete IPv4. El proceso inverso se realiza en la máquina destino, que recibe un paquete IPv6.

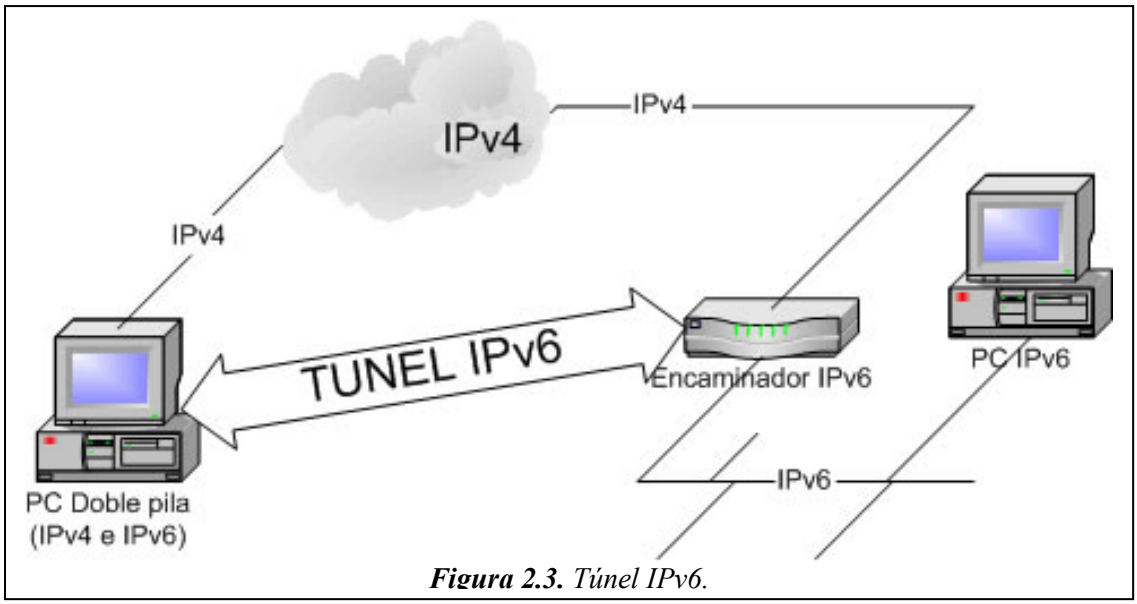


Figura 2.3. *Túnel IPv6.*

DNS en IPv6

Existen dos tipos de registros de DNS para IPv6. El IETF ha declarado los registros A6 y CNAME como registros para uso experimental. Los registros de tipo AAAA son hasta ahora los únicos estándares.

La utilización de registros de tipo AAAA es muy sencilla. Se asocia el nombre de la máquina con la dirección IPv6 de la siguiente forma: NOMBRE_DE_LA_MAQUINA AAAA MIDIRECCION_IPv6

De igual forma que en IPv4 se utilizan los registros de tipo A. En caso de no poder administrar su propia zona de DNS se puede pedir esta configuración a su proveedor de servicios. Las versiones actuales de bind (versiones 8.3 y 9) y el "port" dns/djbdns (con el parche de IPv6 correspondiente) soportan los registros de tipo AAAA.

El tema de IPv6 no es nada nuevo, hace varios años se viene hablando de esta evolución, pero el proceso es algo que vale la pena discutir, enriquecer con noticias, comentarios sobre el mismo y conocer la perspectiva de los usuarios con respecto a la evolución hacia el IPv6.

2.5. Soporte de movilidad

La principal ventaja de los sistemas de comunicaciones inalámbricos móviles es el soporte d ela movilidad, lo que libera al usuario de restricciones de estar situado en un punto concreto. Los sistemas celulares tales como GS/GPRS y UMTS soportan movilidad a través de procedimientos de handover y roaming. El handover se aplica cuando un usuario se mueve a través de las áreas de cobertura de varios nodos en una red inalámbrica y cruza los límites de las áreas. Para soportar handover, los sistemas celulares dependen de sistemas de señalización dedicados que operan en paralelo a la transmisión de contenidos. En los sistemas celulares, el handover entre nodos inalámbricos del mismo tipo, se denomina a menudo Horizontal Handover, y el handover entre nodos inalámbricos de diferentes tipos de redes (GPRS y UMTS) como Vertical Handover. El Roaming puede verse como un caso especial de Handover que requiere acuerdos de control del tráfico entre los operadores y proveedores de red a través de los límites de las regiones (normalmente países).

Las redes WLAN, WMAN y WPAN fueron diseñadas para terminales portables, a menudo en una configuración single-cell. Cumplen las especificaciones para la capa física y la capa de enlace de datos del modelo OSI. Estos sistemas pueden manejar terminales móviles pero con ciertas restricciones. Por ejemplo, en el standard IEEE 802.11, se da soporte a la movilidad de los terminales dentro de la subcapa MAC, lo que implica que el dispositivo puede moverse, pero el mantenimiento de las conexiones establecidas por las capas superiores no está garantizado si los terminales se mueven entre diferentes segmentos LAN. De esta forma, es necesario controlar la movilidad en las capas más altas del modelo OSI. Puesto que las redes ad hoc están diseñadas teniendo en cuenta la optimización de los costes y la simplicidad, se usa por completo el protocolo IP.Esto implica la búsqueda de una solución basada en IP para soportar la movilidad en redes ad hoc. Sin embargo, el protocolo IP no fue diseñado para trabajar en escenarios con movilidad presente, por lo que aún quedan varios problemas por resolver para que los usuarios puedan explotar con garantías todas las posibilidades que ofrecen las redes inalámbricas móviles. Para solucionar estos problemas existen a día de hoy dos enfoques fundamentales, la Mobile IP que describiremos a continuación, o el diseño de protocolos de enrutamiento eficientes adaptados a redes ad hoc que trataremos a lo largo de la tesis con más detalle, para exponer, finalmente, nuestra propuesta.

Antes de pasar a definir Mobile IP, vamos a comentar brevemente algunos conceptos de movilidad y dde como es observada la movilidad desde el punto de vista de la red.

2.5.1. Movilidad

2.5.1.1. Movilidad frente a nomacidad

Los principios del protocolo IP se establecieron en los años sesenta, suponían entre otras cosas que un ordenador es algo grande y pesado y por supuesto, fijo. Desde los años noventa los ordenadores pueden moverse, el ejemplo clásico es el portátil que llevamos de casa al trabajo y del trabajo al hotel. De acuerdo con el diseño del protocolo IP, en cada nuevo punto donde estemos necesitamos una nueva dirección del rango adecuado. El mecanismo m´as habitual para obtener esta dirección es DHCP, aunque en ocasiones tampoco es raro hacerlo manualmente.

El cambio de dirección IP obliga a reiniciar las conexiones, y con ello tal vez las aplicaciones. Esto basta a la mayor parte de los usuarios; es la nomadicidad, o portabilidad como lo denominan otros autores. La aparición de la tecnología inálambrica supone un paso adelante muy importante: los ordenadores pueden tener conexión incluso mientras se mueven. Si se desea mantener la conexión manteniendo la dirección IP, aparece entonces la movilidad, que podemos definir como la habilidad de un nodo para cambiar su punto de conexión a la red desde un enlace hasta otro, manteniendo todas las comunicaciones y la dirección IP.

Cuando un ordenador es estático, no hay ningún inconveniente en que una dirección IP sea al tiempo:

- Un identificador, que distingue un equipo de otro.
- Un localizador, que permite encaminar los datagramas hasta la red donde se encuentra la estación.

Si el ordenador es capaz de moverse, resulta evidente que ambos conceptos deben separarse. A partir del primero, es necesaria una entidad que obtenga el segundo en cada momento. Esta entidad se denomina directorio de localizadores (Location Directory o Mobility Binder). Todo esto es la base de la movilidad, que se puede abordar de diferentes formas. La solución más extendida es Mobile IP.

2.5.1.1.1. Limitaciones de la nomacidad

Disponer de nomadicidad sin movilidad y tener por tanto que cambiar la dirección IP cada vez que la máquina cambie su localización puede no parecer un problema tan serio. En ocasiones la nomadicidad es suficiente, el ejemplo más claro son muchos de los servicios basados en web, donde la dirección IP del navegador web no es importante.

En otros casos, mantener la dirección IP sí es relevante, algunos son evidentes, como mantener conexiones en algunas aplicaciones o permanecer accesibles en un dirección conocida. Pero podemos citar otras situaciones no tan obvias como:

- Aplicaciones cliente/servidor basadas en la dirección IP, por la que se identifique a los clientes.
- La posibilidad de que los servidores sean móviles.
- Software propietario que en su licencia limite el uso según la dirección IP.
- Firewalls u otros elementos de seguridad que impongan el uso de determinado rango de direcciones.

La dificultad que conlleva la administración de un conjunto de direcciones mediante técnicas como DHCP.

2.5.1.1.2. La movilidad y la pila de protocolos

Una cuestión importante es decidir en qué nivel de la pila de protocolos se introduce la movilidad: la mayor´ıa de las veces se hace en el nivel de red, pero podría ubicarse en otros niveles, superiores o inferiores. Hacerlo en nivel de enlace puede tener alguna ventaja (fundamentalmente la facilidad para poner otros protocolos de red por encima) pero también presenta problemas intrínsecos:

- Permite movilidad solo sobre un cierto tipo de medio, pero no la movilidad heterogénea, como por ejemplo pasar de IEEE 802.11 a UMTS.
- El área geográfica es limitada, por la propia naturaleza del nivel de enlace.

Por ello, lo más habitual en desarrollar la movilidad en el nivel de red.

2.5.1.1.3. Macromoviliad y micromovilidad

La división jerárquica de la red en dominios divide a su vez el problema de la movilidad en dos sub-problemas: macromovilidad y micromovilidad.

La macromovilidad trata el movimiento de estaciones entre dominios distintos, típicamente pertenecientes a distintas entidades y a distancias relativamente altas. En micromovilidad el movimiento es dentro de un mismo dominio, lo que suele llevar consigo, reciprocamente, distancias menores (en términos físicos o de latencia) así como una única organización.

Es frecuente combinar protocolos de ambos tipos: un mecanismo de macromovilidad sobre otro responsable de la micromovilidad. Las soluciones más extendidas para macromovilidad son Mobile IP y Mobile IPv6, apenas se consideran otras alternativas. Para micromovilidad el n´umero de propuestas es mayor, aunque todas tienen un funcionamiento muy similar.

2.5.2. Mobile IP

Para que una red Ad-Hoc pueda intercambiar tráfico con Internet es necesario establecer los mecanismos de descubrimiento de pasarelas, las correspondencias entre

las direcciones de las redes Ad-Hoc (no jerárquicas) con las direcciones IP (jerárquicas) y el modo de encaminar.

Esto tiene mucho en com´un con los problemas que resuelven Mobile IP y Mobile IPv6, que trataremos a continuación. Mobile IP (IP móvil) es una modificación de IP para permitir macromovilidad (Figura). El objetivo es enviar datagramas de forma continua a una estación, el nodo móvil (Mobile Node, MN) que se mueve por la red. Con tal de que siga teniendo acceso en el nivel de enlace con algún punto de conexión, el MN puede cambiar su ubicación física en la red manteniendo la misma dirección IP. Esto le permite seguir comunic´andose de forma ininterrumpida con otros hosts en Internet, los denominados Correspondent Nodes (CN).

El funcionamiento de mobile IPv4 se puede describir como tres conjuntos de tareas. Supongamos una estación que sale de su red (Home Network, HN) y llega a una distinta (Foreing Network, FN):

- **Descubrimiento de Agente:** El MN llega a la foreing network, donde descubre un router denominado Foreing Agent (FA).
- **Registro:** El FA asigna al MN una dirección IP válida en su ubicación actual, lo que se denomina care-off address. El MN comunicará esta a su Home Agent (HA). El HA es un router en la Home Network del MN que cooperará con el FA para que el MN reciba los datagramas que vayan dirigidos a su dirección original. Este mecanismo de registro se protege contra usuarios maliciosos con un sistema de autentificación, que por defecto está basado en MD5.
- **Tunneling:** El HA crea un túnel a la Care-Off address del MN.

Los datagramas pueden llegar desde el HA hasta el MN de dos formas:

- Túnel hasta el FA. El HA toma los datagramas, los lleva en un túnel al FA que los desencapsula y envía al MN. El MN sigue usando su home address.
- Túnel hasta el MN. El túnel llega hasta el propio MN. El MN necesita una dirección real de la Foreing Network, lo que se denomina colocated care-off address. Típicamente la habrá obtenido por DHCP.

En caso de que el MN esté en su propia red, en la fase de descubrimiento de agente el MN descubre su Home Agent, en la fase de registro se hace saber al HA que la dirección actual es la IP en la Home Network, con lo que pueden deshacerse los túneles que se hubieran levantado previamente.

2.5.2.1. Mobile IPv6

Todo lo referido sobre Mobile IPv4 es aplicable a Mobile IPv6 (IPv6 Móvil). Se trata de conseguir que un ordenador en una red que no es la suya pueda enviar y recibir datagramas como si no hubiera cambiado su localización física. La aproximación seguida en Mobile IPv6 es muy similar, con la misma terminología y esquemas de funcionamiento. Parte de la funcionalidad específica de Mobile IP desaparece en Mobile IPv6, al estar ya integrada en IPv6. Es el caso de la autentificación y de los Foreing Agents. En IPv4 estos eran necesarios para facilitar a los Mobile Nodes las co-located care-of addreses, de forma que la dirección IP de un mismo FA se empleaba para dirigir tráfico a múltiples MNs. Pero en IPv6 resulta muy fácil facilitar a las estaciones

visitantes una dirección válida en esa red, hay muchísimas disponibles. Además IPv6 cuenta con el descubrimiento de routers proporcionado por el protocolo NDP.

El mecanismo por el que un nodo mantiene su direcci´on de red original en Mobile IPv6 es el siguiente:

- En primer lugar el MN comprueba si está en su Home Network. Mediante una llamada ICMPv6 de descubrimiento de hosts. En caso afirmativo, el proceso concluye.

- Si el MN no está en su HN, obtiene una care-of address, una dirección IP válida en la nueva red. Lo puede hacer de dos formas, o con DHCPv6 o con Stateless Address Autoconfiguration, que consiste en preparar sobre la marcha una dirección concatenando un prefijo de esa red con la dirección de nivel de enlace.

- El MN le comunica a su HA su nueva dirección.

- El HA puede tener constancia de que algunos CN (Correpondent Node) intercambian tráfico con el MN, a estos les hará llegar la nueva dirección. Naturalmente la dirección no puede comunicarse a todos los CN posibles, esto sería tanto como propagarla a todo Internet.

- Los CN que conocen la nueva co-located care-off address envían directamente allí los datagramas, usando una cabecera de encaminamiento IPv6 (routing header) donde la nueva direcci´on es la pen´ultima. Los CN que no conozcan la dirección, envían los datagramas al HA, que los hará llegar al CN por un t´unel.

- Los datagramas que vuelven desde el MN al CN pueden encaminarse sin ningún mecanismo especial, aunque tambi´en se puede usar un túnel.

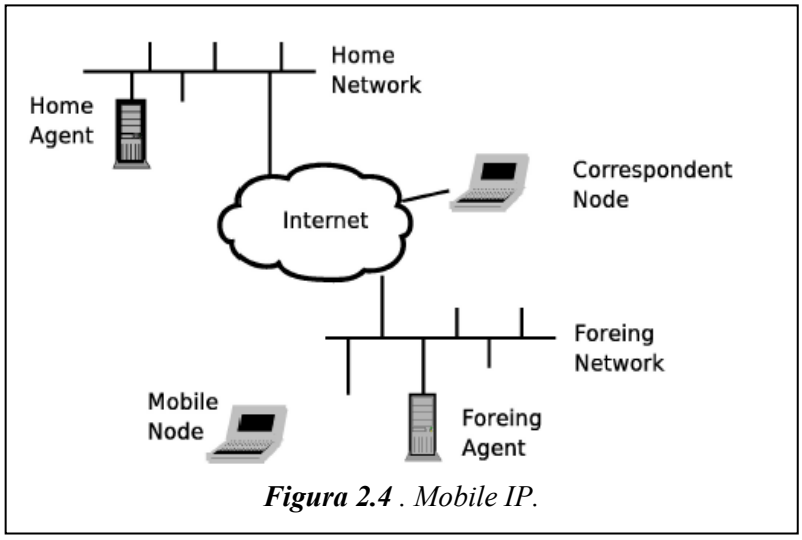


Figura 2.4 *. Mobile IP.*

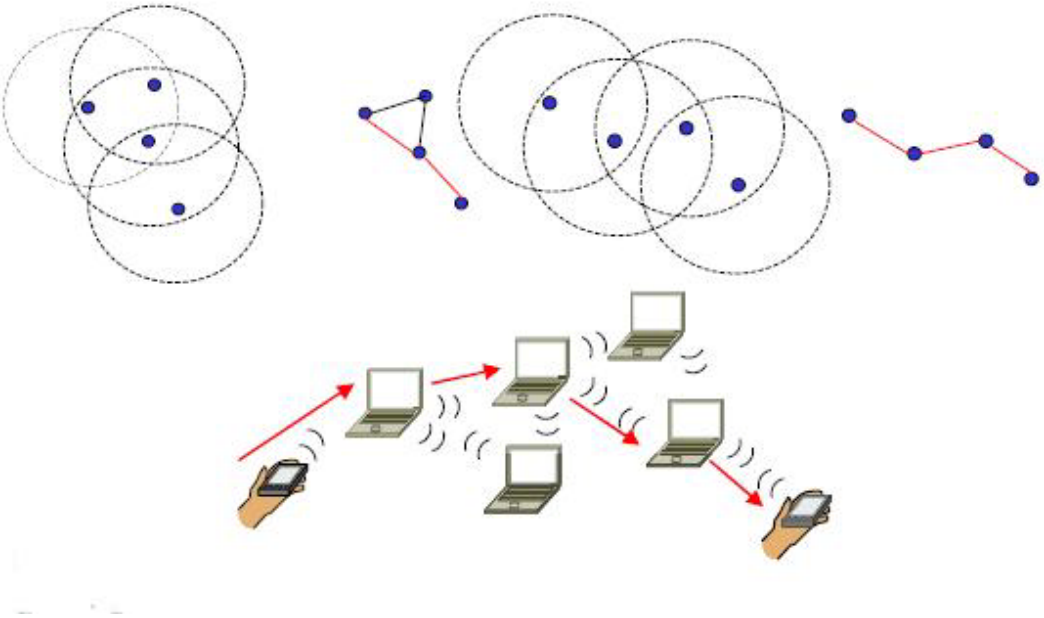
II. REDES AD HOC

1. DEFINICIÓN Y MODELADO DE REDES AD HOC

Una red ad hoc móvil (Mobile Ad hoc NETwork, MANET) consiste en un conjunto de routers libremente conectados. Una MANET se caracteriza por la inclusión de uno o varios interfaces MANET; estos interfaces se caracterizan por la potencial variabilidad de su alcance, siendo asimétricos entre nodos vecinos. Estos routers organizan y mantienen una estructura de enrutamiento entre ellos. Estos routers pueden comunicarse a través de canales inalámbricos dinámicos con alcance asimétrico, pueden ser móviles y pueden unirse y salir de la red en cualquier momento. Estas características dan lugar a temas de estudio en diferentes áreas que analizaremos a lo largo del presente documento.

1.1.Características

Dentro de las redes inalámbricas establecemos 2 grandes grupos separados por una diferencia muy importante, la existencia de lo que conocemos comúnmente por un punto de acceso. Es decir, por un lado disponemos de una red con distintos nodos móviles que se pueden conectar a un nodo central o punto fijo el cual va a realizar funciones más importantes que el resto de nodos. Por el contrario, en el otro grupo no existe dicho punto de acceso, son las redes conocidas como Ad-Hoc.



Por tanto, mientras, en el primer grupo, la comunicación se puede realizar a través de dicho punto de acceso, la comunicación en el segundo grupo se realiza de forma punto a punto, es decir, de un nodo móvil a otro nodo móvil (nodo). De esta forma, en las Ad Hoc cada nodo tiene que actuar tanto de emisor como de receptor para que la comunicación entre dos nodos a través de la red se pueda realizar sin ningún tipo de problema.

Hemos de tener en cuanta también que estas redes se forman cuando se necesitan, que no requieren ninguna infraestructura previa y que todos sus enlaces son inalámbricos. Por supuesto, al ser nodos móviles, no podemos olvidar en ningún momento que son aparatos cuya alimentación viene dada por el uso de baterías, en ningún momento podremos suponer que el nodo está conectado a la red. También, aprovechamos para mencionar, que salvo que se diga lo contrario, todos los nodos están dispuestos a cooperar unos con los otros, es decir, todos están dispuestos a retransmitir, encaminar, producir y consumir mensajes en tanto en cuanto que pertenezcan a la red Ad Hoc y tengan batería suficiente para hacerlo.

Ya hemos comentado que en las redes Ad Hoc los nodos que la forman están dotados de movilidad, esto implica que, generalmente, no exista una topología fija, más bien al contrario, partimos de la base que la topología va a ser cambiante, es decir, en cualquier momento un nodo se puede desplazar de una parte de la red a otra, cambiando, por tanto, los enlaces que disponía con unos nodos de la red por otros nuevos. Es la principal característica a tener en cuenta, ya que la complicación de un diseño de un protocolo eficiente se basa principalmente en la movilidad de los nodos.

También, hemos comentado que, al contrario que otro tipo de redes inalámbricas, estos nodos no se conectan a puntos de acceso, si no que se conectan directamente a otros nodos. Esto supone que cada nodo disponga de una alta conectividad, ya que puede estar conectado a uno o a varios nodos a la vez.

La retransmisión de datos y de paquetes de control se hace sobre un medio compartido empleando ondas electromagnéticas, éste tipo de ondas conllevan una disipación de la señal con la distancia recorrida. Hablar de la disipación de la señal es lo mismo que comentar que cada nodo dispone de un área de retransmisión limitada. Así mismo, existe la posibilidad de colisión de las ondas con distintos objetos del medio o, simplemente, con otras ondas (recordemos que es un medio compartido) Estas colisiones producen interferencias en la señal, las cuales hacen que la tasa de errores sea más elevada.

Respecto al empleo de un medio compartido tenemos que añadir dos características vitales de las redes Ad Hoc. La primera es que el ancho de banda del canal es limitado. Este punto es muy importante a la hora de diseñar un protocolo ya que el envío de paquetes de control que realice cada nodo no debe de ser elevado para no desaprovechar aún más el ancho de banda, reducido, si cabe, por las interferencias ya comentadas. Adelantamos, que el empleo de cabeceras de los paquetes reducidas ayuda a aprovechar el ancho de banda del mismo. La segunda, es que al ser un medio compartido nodos ajenos a la red pueden tener acceso a la información y a los paquetes que circulan por el medio. El tema de la seguridad será tratado más ampliamente en capítulos posteriores a lo largo de esta memoria es que cada nodo solo puede emplear

transmisión Half-Duplex, debido a que una transmisión se colapsa porque su propia emisión genera interferencias.

Como dijimos anteriormente, los nodos disponen de un sistema de alimentación limitado a través de las baterías, lo cual hace que los nodos puedan perder área de retransmisión de la señal, o desconectarse de la red por no tener energía suficiente para permanecer conectado a la misma. Es importante que las operaciones que realicen los nodos en el tratamiento de paquetes sean de poco consumo, para maximizar el ahorro de energía.

Todas estas características suponen que los algoritmos de encaminamiento convencionales para Internet o redes LAN no sirvan. Dicho de forma coloquial, protocolos del estilo de RIP, OSPF o BGP no están pensados para tanto stress. Además, hay que tener en cuenta la posibilidad de que existan nodos unidireccionales en una red. Un nodo B es unidireccional si desde un nodo A puedo enviarle un paquete y que B lo reciba correctamente, pero desde el nodo B no puedo mandar un paquete directamente hasta A. En caso de que el envío desde B hasta A se pudiera hacer hablaríamos de nodos bidireccionales.

Debido a la movilidad el enrutamiento de paquetes tiene que ser calculado de forma dinámica, ya que el desplazamiento de un nodo por la red implica cambios en la ruta por la que deben viajar los paquetes. De la misma forma, dicha movilidad puede implicar la partición de la red en redes distintas en cualquier instante de tiempo. Este último aspecto está relacionado más con protocolos de autoconfiguración más que con protocolos de enrutamiento.

1.2. Arquitectura

1.2.1. Capa física

Los nodos pertenecientes a una red inalámbrica se comunican entre si utilizando como medio de transmisión el espacio libre, es decir, se comunican mediante canales de radiofrecuencia (RF). Los canales de radiofrecuencias presentan características que dificultad la calidad de servicio en redes inalámbricas, algunas de los efectos que sufre la señal cuando se transmite en un canal RF son: el efecto *doopler*, la atenuación de la señal, el desvanecimiento por multitrayecto, etc. A la fecha, lo más común, es que los nodos de las redes ad hoc cuenten con antenas omnidireccionales que les permite comunicarse con cualquier otro dispositivo dentro de su rango de cobertura, el uso de este tipo de antenas influye en la velocidad de transmisión de las redes ah doc. Cuando un nodo desea transmitir o recibir, los nodos vecinos deben de estar en silencio haciendo que la capacidad de la red no se ocupe al 100%. Una de las soluciones propuestas en la literatura para solventar el problema de nodos vecinos es el uso de antenas unidireccionales.

Los estándares IEEE 802.11x definen interfaces para canales de RF en las bandas de los 2.4GHz y de los 5GHz, siendo la primera la más extendida. Esta banda esta muy saturada debido a que también es utilizada por otro tipo de dispositivos y redes como lo son, los hornos de microondas y las redes Bluetooth.

Todo lo anterior da como resultado que los canales de RF utilizados por las redes ad hoc son poco fiables.

1.2.2. Capa de acceso al medio

En general, existen básicamente dos categorías principales de protocolos de control de acceso al medio: los protocolos de acceso aleatorio, en los cuales los nodos compiten entre sí para ganar el acceso al medio de transmisión compartido, y los protocolos de acceso controlado, en los cuales un nodo maestro o de infraestructura decide cuál de los nodos puede acceder al medio de transmisión en cada momento. La falta de una infraestructura y la naturaleza *peer-to-peer* de las redes ad hoc hacen que los protocolos de acceso aleatorio sean la opción natural para el control del acceso al medio en redes ad hoc.

Algunos ejemplos de los protocolos MAC utilizados en las redes ad hoc son los siguientes: MACA (*Multiple Access with Collision Avoidance*), MACAW (*MACA with Acknowledgment*), MACA-BI (*MACA by Invitation*) y FAMA (*Floor Acquisition Multiple Access*). De entre las diversas propuestas, el Comité IEEE 802.11 eligió a CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), una variante de MACA, como la base para sus estándares, debido a su inherente flexibilidad y porque resuelve los problemas de los terminales oculto y expuesto a través del sencillo mecanismo de intercambio de señales de control RTS/CTSDATA/ACK.

Entre los Protocolos MAC de Acceso Controlado se encuentran los siguientes: TDMA (*Time Division Multiple Access*), FDMA (*Frequency Division Multiple Access*), CDMA (*Code Division Multiple Access*) y TSMA (*Time Spread Multiple Access*).

Aunque estos protocolos raramente se utilizan en redes ad hoc, se prefieren en ambientes que necesitan garantías de Calidad de Servicio (QoS), ya que sus transmisiones están libres de colisiones. Su aplicación está principalmente adaptada a redes como *Bluetooth* y a otras redes ad hoc basadas en grupos (*clusters*) de nodos, en las cuales el acceso al medio está controlado por nodos maestros que deciden qué nodo del grupo es el que tiene acceso al canal, posibilitando así transmisiones libres de contienda y colisión. Entre las propuestas para mejorar el comportamiento de los protocolos MAC para redes ad hoc basadas en protocolos de acceso aleatorio se incluyen, por ejemplo, algoritmos para reducir el consumo de energía de los nodos móviles que permiten que los nodos “duerman” (esto es, que pasen a un estado de bajo consumo de energía) durante el periodo ocioso; diversos algoritmos de gestión de paquetes; reducción de los radios de transmisión y de recepción; ajuste de la velocidad de transmisión entre nodos vecinos dependiendo de la distancia que los separa, y diferentes esquemas de codificación y modulación de señales.

1.2.3. Capa de red

Una característica especialmente importante de los protocolos de encaminamiento para redes ad hoc es que deben poder adaptarse rápidamente a lo cambios continuos de la red, con el fin de mantener las rutas entre los nodos que se están comunicando. De manera general, los protocolos de encaminamiento para redes

ad hoc se clasifican en dos categorías principales: proactivos y reactivos, aunque también existen otras clasificaciones. Los protocolos proactivos mantienen tablas que almacenan la información de encaminamiento y periódicamente, o ante cualquier cambio en la topología de la red, disparan un mecanismo de propagación de actualización a través de la red, con el fin de mantener una idea real del estado de la red. Esto puede causar una cantidad importante de paquetes de señalización (*overhead*) que afecte la utilización del ancho de banda, el caudal (*throughput*), así como el consumo de energía. La ventaja es que las rutas a cada destino están siempre disponibles sin el incremento de paquetes de señalización que ocasiona un mecanismo de descubrimiento de rutas, pero tales protocolos tienen problemas para funcionar adecuadamente cuando en la red se presenta una alta tasa de movilidad o cuando hay un gran número de nodos en la red. Ejemplos de protocolos de esta categoría son: DSDV (*Destination-Sequenced Distance-Vector*), WRP (*Wireless Routing Protocol*), CGSR (*Cluster Gateway Switch Routing*), FSRP (*Fisheye State Routing Protocol*), OLSR (*Optimizad Link-State Routing*) y TBRPF (*Topology Dissemination Based on Reverse-Path Forwarding*). Los últimos dos protocolos preactivos se han convertido en RFC´s (*Requests For Comments*) experimentales aceptados por la IETF.

Por otro lado, los protocolos de encaminamiento por demanda o reactivos se caracterizan por iniciar un mecanismo de descubrimiento de ruta cuando una fuente necesita comunicarse con un destino al cual no sabe cómo llegar. El descubrimiento de ruta se realiza normalmente mediante una inundación de la petición. De manera general, el encaminamiento por demanda requiere menos *overhead* que el encaminamiento basado en tablas (proactivo), pero incurre en un retraso de descubrimiento de ruta cada vez que se requiere un nuevo camino.

Las diferencias entre los protocolos por demanda están en la implementación del mecanismo de descubrimiento de ruta y en las optimizaciones del mismo. Ejemplos de protocolos reactivos son los que se mencionan a continuación: El protocolo DSR (*Dynamic Source Routing*) que está muy cerca de convertirse en RFC experimental. Por su lado, el protocolo AODV (*Ad hoc On- Demand Distance Vector*) ya es un RFC experimental. Otro protocolo reactivo que inició el camino de convertirse en RFC pero que ya no ha tenido avances en ese proceso es TORA (*Temporally Ordered Routing Algorithm*). El protocolo de encaminamiento DYMO (*Dynamic MANET On-demand*) es el último protocolo reactivo unicast que está siendo considerado por el MANET-WG para convertirse en RFC, pero se encuentra apenas en sus primeras fases para lograrlo.

Además de los protocolos proactivos y reactivos están los protocolos híbridos. El protocolo ZRP (*Zone-Based Hierarchical Link State Routing Protocol*) es un ejemplo de protocolo híbrido que combina los enfoques proactivo y reactivo, tratando de combinar las ventajas de ambos.

1.2.4. Capa de transporte

Las redes MANET tienen como objetivo final la implantación de la Internet inalámbrica omnipresente y ubicua, por lo que está basada en la pila de protocolos TCP/IP. Esto ha significado que prácticamente todos los esfuerzos de investigación y desarrollo del MANET-WG estén principalmente dirigidos al desarrollo de algoritmos

de encaminamiento especialmente adecuados para este tipo de redes, sin que los demás protocolos de la pila se vean afectados. Aun así, en los últimos años se han presentado varias propuestas encaminadas a mejorar las prestaciones de los principales protocolos de transporte de la Internet, TCP (*Transmisión Control Protocol*) y UDP (*User Datagram Protocol*), en entornos ad hoc. Entre estas mejoras está el emplear mecanismos de señalización explícita que permitan tener un TCP "amigable" con las redes ad hoc, para que no active, erróneamente, el mecanismo de control de congestión ante la pérdida de la ruta entre el nodo fuente y el destino final debida a un fallo en el enlace. Otros autores han propuesto la creación de TPA (*Transport Protocol for Ad hoc Networks*), un nuevo protocolo de transporte especialmente diseñado para entornos ad hoc, que incluye mecanismos para detección de pérdida del enlace y recuperación de ruta, además de que establece un mecanismo de control de congestión diferente al de TCP. Los protocolos de transporte SCTP (*Stream Control Transfer Protocol*) y el MRTP (*Multiflow Realtime Transport Protocol*) han sido especialmente diseñados para ofrecer un mejor transporte de datos a las aplicaciones de tipo media-streaming y de tiempo real.

1.2.5. Capa de aplicación

Uno de los objetivos del MANET-WG es que las mismas aplicaciones diseñadas para la Internet actual puedan funcionar en las MANETs. Esto es válido actualmente para las aplicaciones de transmisión de datos que pueden conformarse con una entrega de datos bajo la filosofía de *Best Effort*, sin grandes requerimientos de ancho de banda ni restricciones en cuanto al tiempo de entrega de los paquetes. Sin embargo, Internet también se utiliza actualmente para otros tipos de aplicaciones, como aquellas que permiten el acceso en tiempo real a contenidos de audio y vídeo en repositorios remotos que producen flujos continuos de datos (*streams*), así como aplicaciones multimedia interactivas de tiempo real, como la voz sobre IP (*Video over Internet Protocol, VoIP*) o telefonía IP (*IP Telephony*), la vídeo conferencia y la vídeo telefonía, que demandan a la red de transporte ciertas garantías mínimas de retardo en la entrega de paquetes, de variación de este retardo y de ancho de banda, pero que por otro lado pueden soportar cierto nivel de pérdida de paquetes, sin que por tanto la información se vuelva inútil.

Debido a que el ancho de banda y la fiabilidad de los enlaces de las redes ad hoc es mucho menor que en las redes cableadas, hay muchos retos por resolver para que este tipo de aplicaciones puedan ser soportadas de manera adecuada en los entornos ad hoc. Algunas propuestas van en el sentido de disminuir lo más posible el ancho de banda requerido por las aplicaciones, las cuales se basan en nuevas técnicas de codificación y compresión de la información, sacrificando la calidad de la información para todos los nodos y condiciones de la red por igual. Otra dirección que se está considerando cada vez más, es hacer que las aplicaciones mismas se adapten a las condiciones dinámicas de las redes ad hoc y a las capacidades particulares de los nodos comunicantes en ambos extremos.

1.3.Tipos

1.3.1. Bluetooth

La tecnología Bluetooth es un estándar y un medio para la conexión sin cables entre el teléfono y otros dispositivos electrónicos. La tecnología fue desarrollada por Ericsson y implementada por Nokia, Ericsson, IBM, Intel y Toshiba. Mientras tanto, el grupo de trabajo del Bluetooth ha sido enriquecido con centenares de representantes de empresas como, por ejemplo, One2One, Motorola, Qualcomm, Compaq, Dell, 3Com Palm, VLSI, Xircom, Psion Dacom y Lucent.

Esta tecnología se asemeja a la de las redes de telecomunicaciones pero el espectro radioeléctrico que ocupa es libre y no está sometido a licenciamiento (en la banda de los 2.45 gigahertz). La velocidad de transmisión de datos con el Bluetooth deberá situarse entre los 720 kbps y un megabit por segundo (Mbps).

El Bluetooth va a potenciar la existencia de LAN's (Local Area Networks) inalámbricas, en que los distintos terminales de los más variados equipos electrónicos podrán comunicar y trocar información en movimiento, sin que existan hilos y cables entre si. Esto significa que los aparatos podrán ser utilizados sin la necesidad de buscar las versiones más recientes o compatibles.

1.3.2. Redes Mesh

Una red MESH es aquella que emplea uno o dos arreglos de conexión, una topología total o una parcial. En la total, cada nodo es conectado directamente a los otros. En la topología parcial los nodos están conectados solo a algunos de los demás nodos. Esto está mejor ilustrado en una red total simple, como se observa en la Figura 1.1, en la cual todos los nodos (computadores) están conectados a todos los demás.

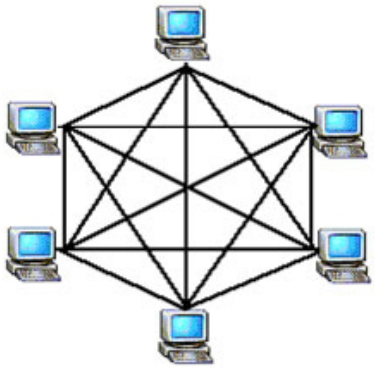


Figura 1.1. *Red mesh completamente conectada.*

Seguidamente, en la Figura 1.2 podemos observar un diagrama de una red MESH parcial, parecido a una implementación de MESH inalámbrico más realista: Los nodos tienen un grado variable de conexión, con algunos nodos conectados a muchos nodos y otros en los extremos con pocas o una sola conexión.

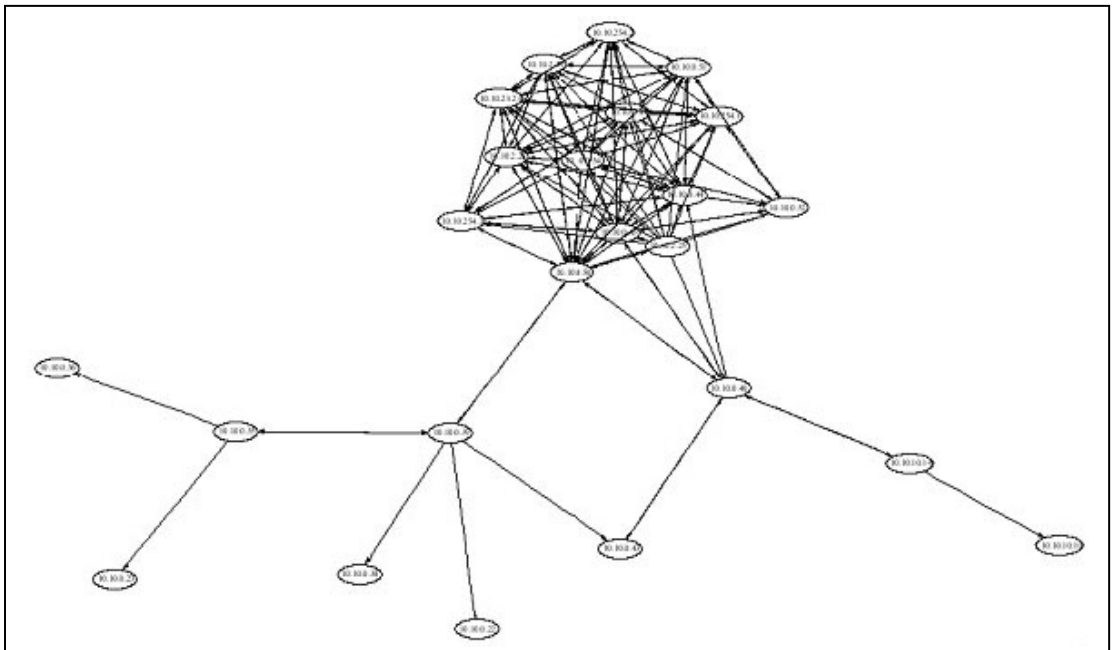


Figura 1.2. *Diagrama de una red mesh parcial.*

Como se puede ver en la definición, nada es necesariamente dinámico en una red MESH. Sin embargo, en años recientes, y en redes de conexión inalámbricas, el termino "MESH" es a menudo usado como un sinónimo de "ad hoc" o red móvil. Obviamente combinando las dos características de la topología MESH y las capacidades de ad hoc, es una proposición muy atractiva.

Mientras algunos ven las grandes ventajas de una red MESH en entornos dinámicos, la mayoría de las implementaciones más relevantes y exitosas que han surgido hasta ahora, son completamente estáticas, como por ejemplo con nodos/antenas colocados en techo. Es útil recordar el entendimiento común y corriente de redes MESH como: "redes que manejan conexiones de redes, todos contra todos, que son capaces de actualizar y optimizar dinámicamente estas conexiones".

Un escenario típico MESH en una zona urbana puede verse así, conectando mayormente antenas en techos. Pero potencialmente incluyendo muchas otras ubicaciones, como torres de antenas, árboles, nodos móviles (vehículos, laptop).

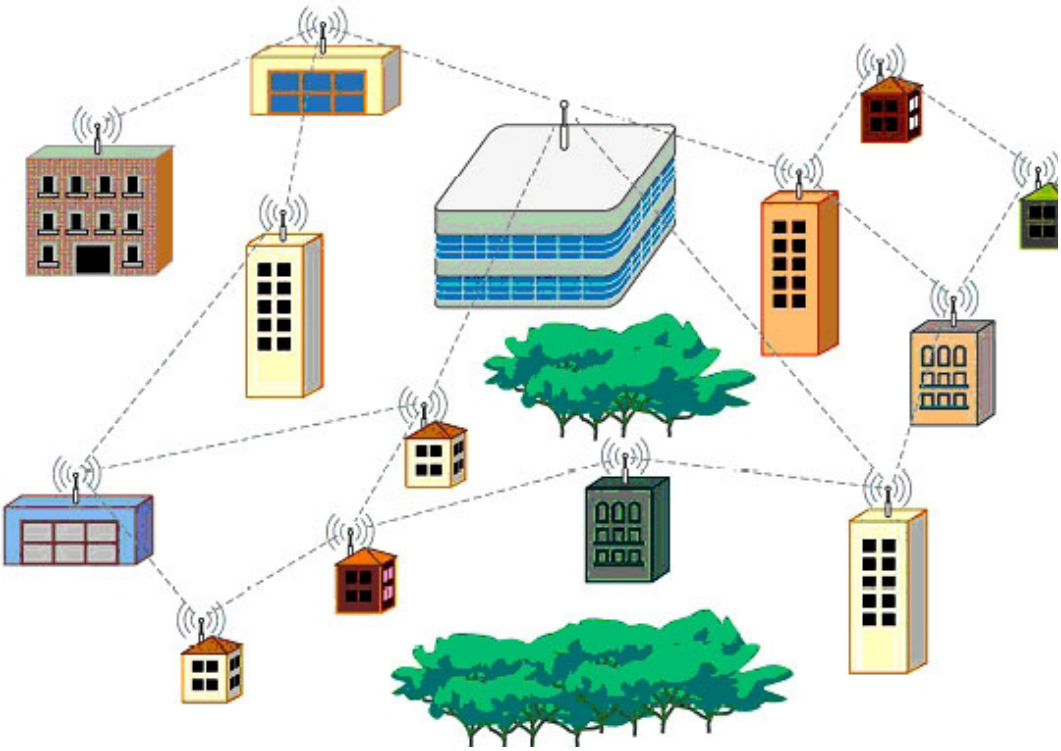


Figura 1.3. *Escenario típico de una red mesh.*

La tecnología de las redes MESH están madurando gradualmente a un punto donde no puede ser ignorada, cuando se considera el despliegue de las tecnologías de redes inalámbricas. El primer despliegue de una comunidad MESH en gran escala (hasta algunos cientos de nodos) han demostrado suficientes ventajas para motivar futuros experimentos.

Estas son algunas de las razones del porque las redes MESH son vistas como una opción atractiva:

- **Ajustes reales.** En la realidad la topografía raramente viene en forma de anillo, línea recta o estrella. En terrenos difíciles, sean remotos, rural o urbano, donde no todos los usuarios ven uno o algunos puntos centrales, lo mas posible es que el usuario solo vea a uno o mas usuarios vecinos.
- **Precio.** El hecho que cada nodo MESH funciona tanto como cliente y como repetidor potencialmente significa ahorro en el número de radios necesarios y por lo tanto en el presupuesto total. Mientras este punto pierde relevancia con la caída de los precios de radios, la cercanía de las redes MESH puede reducir la necesidad de torres centrales (costosas y vulnerables) y otras infraestructuras centralizadas.
- **Organización y modelos de negocio.** La naturaleza descentralizada de las redes MESH se presta muy bien para un modelo de propiedad en donde cada participante de la red posee y mantiene su propio hardware, el cual simplifica significativamente los aspectos financieros y comunales del sistema.
- **Facilidad y simplicidad.** Para un artefacto que esta preinstalado con software de MESH inalámbrico y usa protocolo standard como el 802.11b/g, el montaje es extremadamente simple. Ya que las rutas son configuradas dinámicamente, es

generalmente suficiente arrojar la caja en la red y juntar cualquier antena requerida para alcanzar uno o más nodos vecinos existentes.

- **Red robusta.** Las características de la topología de una red MESH y del enrutamiento AdHoc prometen gran estabilidad en cuanto a condiciones variables o en alguna falla de algún nodo en particular. La cual va a estar bajo duras condiciones experimentales.
- **Potencia.** Los nodos de una red MESH, exceptuando posiblemente aquellos nodos que mantienen un enlace directo con Internet, pueden ser construidos con bajísimos requerimientos de energía, es decir, pueden ser desplegados como unidades completamente autónomas con energía solar, eólica, hidráulica, celdas combustibles (derivados del petróleo) o generada por tracción de sangre.
- **Integración.** El hardware de las MESH tiene todas las ventajas de una tecnología firme y simple: típicamente pequeño, no hace ruido y fácilmente encapsuladas en cajas a prueba de agua. Esto significa que integra agradablemente a la intemperie así como también para usar dentro de los hogares.
- **Entornos urbanos y rurales.** Hasta ahora, las redes MESH han sido mayormente propuestas para redes urbanas y redes municipales. Sin embargo, hay un gran potencial para redes MESH en zonas de conectividad rurales o lejanas.
- **Tópicos y limitaciones.** Como cualquier tecnología existen limitaciones y tópicos para las redes MESH, la mayoría de estos están basados alrededor de los límites del ancho de banda, escalabilidad y las dificultades de garantizar calidad de servicio. Esto será discutido en detalle en su propia unidad. Es importante estar al tanto del hecho que las estructuras organizacionales y comunicacionales de un proyecto no son necesariamente reflejados uno a uno por la estructura técnica de una red. Ellos pertenecen a dominios diferentes.

1.3.3. Redes de sensores

Desde hace algunos años, han comenzado a emerger las Redes de Sensores. Los sensores son fuentes de información tan variados como lo son las medidas que realizan. Los hay de temperatura, de luminosidad, de presión, de humedad, de velocidad, de aceleración, de presencia, de volumen y un sin fin de magnitudes que se nos ocurran. Si a estos sensores que nos reportan información valiosa para nuestras vidas, les añadimos la capacidad de comunicación inalámbrica y la posibilidad de formación de redes *ad-hoc*, obtenemos las Redes de Sensores Inalámbricos, que están teniendo un auge cada vez mayor debido principalmente a la multitud de aplicaciones que se están desarrollando.

Encontramos estas aplicaciones sin más que mirar a nuestro alrededor: la contaminación de una ciudad se mide con unos sensores de polución, de CO2, Ozono, etc que, envían los datos en tiempo real y sin necesidad de ningún tipo de interacción por parte humana a un centro de control, y cuando ésta sobrepasa ciertos niveles, se generan unas alarmas para que se tomen las medidas oportunas .En los edificios existen sensores de presencia, que se conectan con el centro de control y emiten alarmas cuando detectan presencias en momentos en que no debiera haber nadie. También esta información procedente de los sensores puede ser procesada y provocar una acción

correctora. Así por ejemplo en una bodega, con sensores de humedad repartidos por toda la planta, cuando se detecta alguna zona en la que la humedad supera cierto umbral se encienden unos ventiladores o se encienden unos micro-aspersores en caso de defecto de humedad.

En definitiva, las aplicaciones de este tipo de redes son múltiples, desde aplicaciones de seguimiento, de seguridad, de salud, de gestión ...

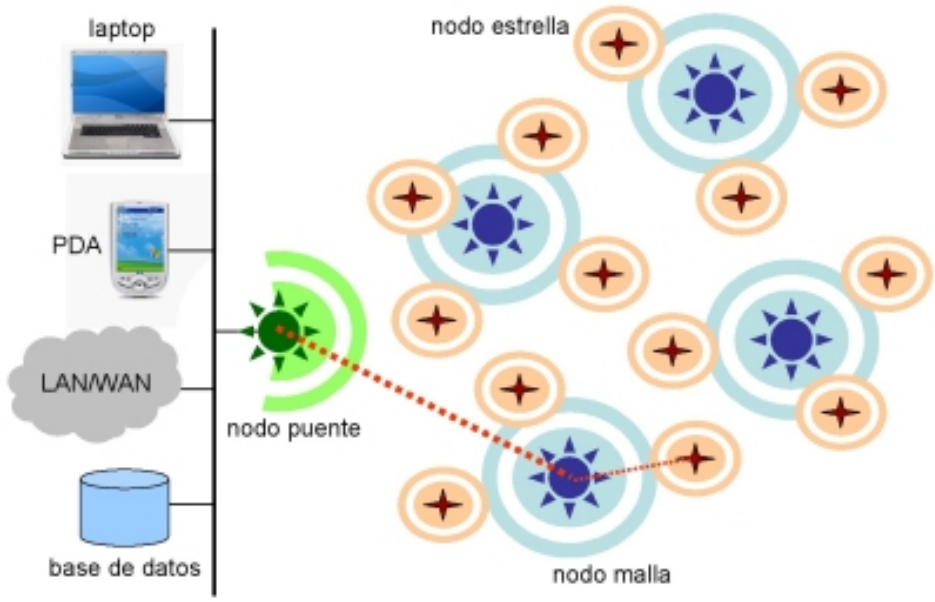


Figura 1.4. *Red de sensores.*

1.3.3.1. Modos de uso de las redes de sensores

Las redes de sensores pueden operar de las siguientes formas:

- Monitorización continua:
  - Nodos midiendo los mismos parámetros en un área de interés. Envío periódico de la información recogida.
  - Aplicación: control de la agricultura, microclimas, etc.
- Monitorización basada en eventos:
  - Nodos monitorizando entornos continuamente. Pero sólo hay envío de información cuando ocurre algún evento.
  - Aplicación: control de edificios inteligentes, detección de incendios, aplicaciones militares, etc.
- Localización y seguimiento:
  - Los nodos se usan para etiquetar y localizar objetos en una zona determinada.
  - Aplicación: rastreo de animales, seguimiento de un trabajador, etc.
- Redes híbridas:

- Escenarios de aplicación que contienen aspectos de las tres categorías anteriores.

1.3.3.2. Aplicaciones de las redes de sensores

- Aplicaciones militares:
  - Monitorización de fuerzas y equipos enemigos.
  - Vigilancia en el campo de batalla.
  - Reconocimiento del terreno.
  - Detección de ataques biológicos, químicos o nucleares.
- Aplicaciones medioambientales:
  - Seguimiento de animales.
  - Monitorización de las condiciones ambientales en cultivos, riego.
  - Agricultura de precisión.
  - Detección de incendios forestales.
  - Detección de inundaciones.
  - Estudios de contaminación.
  - Prevención de desastres.
  - Monitorización de áreas afectadas por desastres, etc.
  - Estudios sísmicos.
  - Seguridad de estructuras.
- Aplicaciones médicas:
  - Telemonitorización de datos fisiológicos en pacientes, diagnóstico.
  - Administración de medicamentos.
  - Seguimiento de médicos y pacientes en hospitales.
- Aplicaciones en el hogar/edificios:
  - Domótica
  - Control de electrodomésticos
  - Entornos inteligentes
  - Control ambiental
- Aplicaciones industriales:
  - Seguimiento de vehículos
  - Control de flota
  - Control de inventarios
- Aplicaciones turísticas:
  - Interactividad en museos y espacios turísticos.
  - Control de acceso.

1.3.3.3. Ejemplos de redes de sensores

1.3.3.3.1. ZigBee

Una vez que en los apartados anteriores hemos hecho una recorrido general por las redes de sensores, sus caracter´ısticas y aplicaciones, a continuación nos centraremos en algunas tecnologías concretas. Comenzaremos por ZigBee, un standard reciente para la normalización de redes de sensores, promovido por un consorcio de empresas: la ZigBee Alliance. Define un sistema completo de redes inalámbricas con baja velocidad de transferencia de datos para dispositivos muy sencillos, muy baratos y de un consumo

tan bajo como para ser capaces de funcionar meses o años sin recargar sus baterías. Para los niveles físico y de enlace, ZigBee confía en el standard de comunicaciones IEEE 802.15.4, al que añade un nivel de red, de seguridad y un marco de trabajo para las aplicaciones (application framework), quedando las aplicaciones y los perfiles de usuario fuera del standard (Figura 1.5).

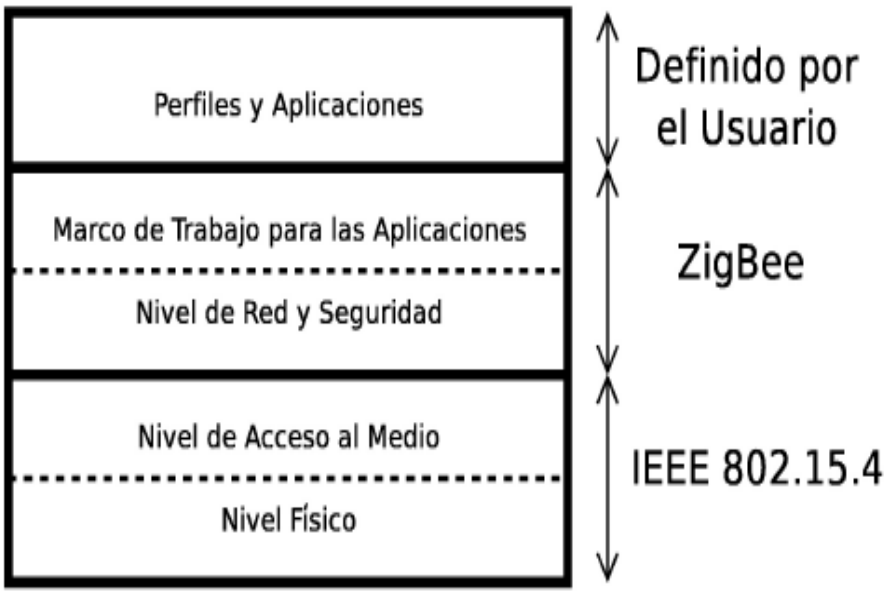


Figura 1.5. *ZigBee y IEEE 802.15.4*

En una red 802.15.4 pueden operar dispositivos de funcionalidad completa (FFD, Full Funcion Device) y dispositivos de funcionalidad reducida (RFD, Reduced Funcion Device). Los FFD pueden comunicarse tanto con FFD como con RFD, mientras que los RFD solo pueden hacerlo con un FFD. El nivel de enlace ofrece topología en estrella, el nivel de red permite la formación de redes Ad-Hoc con un protocolo basado en AODV. Combinando redes Ad-Hoc con configuraciones en estrella resulta la topología en árbol. Estas tres topologías (star, mesh y cluster-tree) ofrecen mucha más versatilidad que por ejemplo BlueTooth, que está limitado una configuración en modo cliente-servidor.

1.3.3.3.2. TinyOS

TinyOS es un sistema operativo orientado a redes inalámbricas de sensores. Creado originalmente por la Universidad de California en Berkeley en 2001, en la actualidad es un proyecto de software libre.

Ofrece un modelo de ejecución guiado por eventos y una arquitectura basada en componentes que incluye protocolos de red, servicios distribuidos, drivers para sensores y herramientas de adquisición de datos. Está portado a una docena de plataformas y a un gran número de sensores, siendo el sistema operativo de este tipo con una mayor comunidad de usuarios.

1.3.3.3.3. Motas

En 2001 fue acuñado el término polvo inteligente (smart dust) para redes de sensores donde cada nodo tendría un volumen de un milímetro cúbico, dimensiones con las que es muy difícil trabajar actualmente. Una plataforma muy extendida hoy, mucho más práctica y acorde con la tecnología actual, son las motas, ligadas a la Universidad de California en Berkeley y su sistema operativo TinyOS.

Nombre	Año	ROM (Kb)	RAM (Kb)	Vel.Trans. (Kbps)	Consumo (mW)
WeC	1998	8	0,5	10	24
René	1999	8	0,5	10	24
René 2	2000	16	1	10	24
Dot	2000	16	1	10	24
Mica	2001	128	4	40	27
Mica2Dot	2002	128	4	38,4	44
Mica 2	2002	128	4	38,4	89
Telos	2004	60	2	250	41

Emplean tecnología ya disponible en el mercado (off the shelf), las más pequeñas son como un reloj de pulsera, las más grandes tienen dimensiones inferiores a las de un paquete de pañuelos de papel. En la Figura podemos ver un resumen de las características de las distintas generaciones de motas. Una mota pasa la mayor parte de su vida dormida, se despierta en ciertas ocasiones, hace su trabajo y vuelve a dormir. Esto le permite funcionar entre 300 y 900 días sin reemplazar sus baterías (2 pilas R6 convencionales).

La última generaci´on es la plataforma Telos que dispone de un conector USB y usa tramas compatibles con IEEE 802.15.4 (aunque no soporta este protocolo por completo). Las arquitecturas anteriores usaban componentes de radio no normalizados. En 2006 el precio una unidad comprada en pequeñas cantidades es de unos 80 dólares.

1.4. Aplicaciones

Es fácil encontrar situaciones donde se ve la utilidad de las redes Ad-Hoc. Uno de los ejemplos más clásicos (aunque también discutido) es una reunión de trabajo: un grupo de personas con ordenadores portátiles o PDAs. Son de distintas empresas y por tanto sus direcciones son distintas. Tal vez en la sala haya acceso a Internet y puedan usar por ejemplo IP móvil, pero ¿para qué pasear sus datagramas por todas la ciudad o todo el país cuando están en la misma habitación? Sus equipos probablemente estén dotados de puertos de infrarrojos o bluetooth que les permitan formar una red para la ocasión. En algunos casos, simplemente no habrá infraestructuras de apoyo. Pensemos en poblaciones aisladas o de orografía difícil, situaciones de emergencia, desastres naturales donde las infraestructuras hayan desparecido, etc.

Otro ejemplo son las denominadas PAN (Personal Area Networks) o piconets: redes formadas por los dispositivos de una persona, como su reloj, su agenda y su teléfono móvil. Una red así puede querer entrar en contacto con la red de otra persona que en ese momento esté próxima.

La capacidad de desplegarse inmediatamente y la no dependencia de un único punto de fallo hace a estas redes muy interesantes para el uso militar, de hecho uno de

los orígenes de esta idea está en la agencia de proyectos de investigación avanzada para la defensa (DARPA, Defense Advanced Research Projects Agency) del ministerio de defensa de Estados Unidos.

El campo militar es posiblemente el más desarrollado actualmente, el ejército estadounidense ya dispone de un sistema basado en este tipo de redes, el FBCB2 (Force XXI Battle Command, Brigade-and-Below). Uno de sus objetivos es distinguir las fuerzas propias de las fuerzas del enemigo, ofreciendo a los soldados una visión del campo de batalla similar a la de un videojuego. Los equipos de la generación inmediatamente anterior estaban basados en comunicaciones por satélite, con latencias de cinco minutos. En abril de 2003 el FBCB2 se utilizó en la segunda guerra del golfo, lo que supuso probablemente el primer uso bajo fuego real de una red Ad-Hoc.

Otro motivo por el que una red Ad-Hoc puede ser ventajosa es el coste.

Aunque exista una infraestructura de red, si pertenece a una entidad ajena es muy posible que nos cobre por su uso, mientras que si tenemos nuestros equipos desplegados dispondremos ya de una red sin coste adicional. Por ejemplo los coches que pasan por una autopista podrían formar fácilmente una red Ad-Hoc, independiente de su capacidad de conectarse a otras redes como GSM, o similar. Por último, supongamos que tenemos estaciones capaces de comunicarse empleando un satélite. Estos equipos de comunicaciones son caros, pero bastaría con que algunos tengan capacidad de conectarse al satélite para que todos dispusieran de conectividad. Y no todos los capaces de conectarse al satélite necesitarían estar conectados simultáneamente.

Aunque se puedan pensar muchos usos, la killer ap (programa de ordenador útil para los usuarios, que provoca un aumento importante en las ventas de cierto hardware o sistema operativo necesario para su funcionamiento) puede ser cualquier otra aplicación que hoy no imaginamos. En todo caso, para nosotros y para muchos especialistas resulta evidente que el potencial de este tipo de redes es muy grande.

1.5. Problemas abiertos en redes Ad Hoc

Los principales aspectos en los que se investiga actualmente en el ´ambito de las redes Ad-Hoc son:

- La escalabilidad, tal vez el principal problema. Hoy con apenas 50 o 100 nodos los resultados empiezan a ser insatisfactorios.
- El ahorro de energía. La batería es un bien escaso y muy preciado, es muy importante buscar formas de optimizar su aprovechamiento. Hay técnicas como dormir nodos para despertarlos en determinados intervalos de tiempo, lo que exige una sincronización no trivial. Algunos sistemas permiten transmitir con más o menos fuerza, modificando el alcance de la transmisión.
- La premisa de la buena fe: las redes Ad-Hoc se basan en la cooperación bienintencionada entre estaciones. En las redes ordinarias hay que tratar los problemas ocasionados por nodos maliciosos, pero en estas redes la cuestión es aún más compleja porque no es necesaria la hostilidad para causar daño, basta la ausencia de altruismo. Por ejemplo, si el nivel de la batería está bajo puede ser

legítimo no gastarlo retransmitiendo para los demás, o limitarse a los mensajes más urgentes. Pero esto puede dar lugar a abusos.

- El modelo cliente-servidor es el habitual en Internet, pero no es adecuado en redes Ad-Hoc, ya no hay entidades bien conocidas que ofrezcan servicios. Precisamente una pregunta relevante es ¿dónde están los servicios?, lo que resulta difícil en entornos tan cambiantes. Los servicios pueden buscarse al mismo tiempo que las rutas, pero esto va contra el modelo de capas, cuya bondad está demostrada. Una alternativa es basarse en direcciones multicast bien conocidas, esto parece adecuado para servicios básicos como DNS o DHCP, aunque en redes Ad-Hoc es un tema abierto.
- La seguridad: lo que está en el aire puede ser capturado fácilmente. La solución es cifrar las comunicaciones, pero la seguridad y el cifrado se basan en una distribución segura de claves y en una autoridad certificadora centralizada, y esto último es casi una contradicción en términos tratándose de redes Ad-Hoc. Además, el ancho de banda y la capacidad de procesamiento requerido para hacer seguras las comunicaciones pueden suponer una carga muy importante.
- El multicast (multidifusión), que puede ahorrar drásticamente ancho de banda en distribución masiva. No est´a claro si debe incluirse en este nivel de los algoritmos. Por un lado, los problemas de multidifusión pueden ser de naturaleza lo bastante específica como para que no merezca la pena mezclarlos con otras cuestiones. Pero por otra parte, los protocolos dedican mucho esfuerzo a conocer el estado de nodos intermedios que cambian continuamente. Si el multicast se convierte en una acumulación de unicast, muy posiblemente se estarán desperdiciando recursos.
- La simetría en los enlaces: si la estación A puede transmitir a la estación B, lo más habitual es que B pueda transmitir a la estación A. Pero en algunas tecnologías concretas no sucede esto, lo que complica notablemente todos los algoritmos.
- La gran variedad de medios físicos distintos, todos incompatibles entre sí. Cuando todos los nodos emplean la misma tecnología en el nivel físico, pueden trabajar juntos, pero solo en ese caso. Esto es un serio inconveniente para llegar a un protocolo standard.
- La ubicación en la torre de protocolos. Normalmente se hace en nivel de red. Esto presenta circunstancia favorables: en las tablas de encaminamiento se trabaja con direcciones de red que las aplicaciones resuelven en direcciones de enlace: en la dirección de enlace del propio nodo destino si es accesible directamente, o bien la dirección de enlace del nodo que hace de router y que se lo hará llegar. Si la movilidad está en el nivel de enlace, entre otras cosas hay que resolver direcciones de enlace con otras direcciones de enlace, lo que resulta poco natural.
- La calidad de servicio: podemos indicar si los enlaces son válidos o no, pero apenas hay expresividad para indicar cambios en su calidad.
- La adaptación de las aplicaciones al ancho de banda disponible. Usando enlaces inalámbricos, el ancho de banda es típicamente un orden de magnitud inferior al cable. Sería deseable que las aplicaciones fueran conscientes de ello, y en cada caso adapten la información enviada: tal vez reduciendo el número de píxeles por segundo si se trata de vídeo, el número de colores o la resolución en las imágenes, u omitiendo en páginas web animaciones de utilidad dudosa.
- Las máquinas de recursos limitados: este problema obliga a que nuestros mecanismos sean ligeros, para poder ejecutarse en máquines con una capacidad

computacional limitada, no necesiten almacenar mucha información debido a la escasez de memoria, y no consuman demasiada energía para aumentar la duración de las baterías. El tema de la eficiencia energética de las comunicaciones en redes ad hoc ha dado pie a numerosas investigaciones y publicaciones. Sin embargo, en esta tesis no hemos abordado este aspecto que, sin embargo, desarrollaremos en próximos trabajos.

2. Calidad de servicio en redes ad hoc (QoS)

La necesidad de que Internet soporte QoS se hace patente viendo el incremento de la actividad investigadora del IETF para dar soporte a la arquitectura Diffserv. Los diseñadores iniciales de Internet se alejaron del diseño de la red telefónica donde la inteligencia se encontraba en la red mientras los terminales carecían prácticamente de ella. La red telefónica proporciona QoS en el sentido de que garantiza conexión y calidad de voz una vez se establece dicha conexión. La idea inicial de Internet de crear una red simple moviendo la inteligencia a los enlaces y los terminales finales, colaboró en el rápido crecimiento de Internet. Pero con la proliferación de aplicaciones que requieren algún tipo de garantía de servicio por parte de la red, convierte en esencial el soporte de QoS por parte de Internet. Las comunicaciones multimedia y VoIP (Voide over IP) son dos ejemplos de aplicaciones que están ganando rápidamente popularidad. El éxito de estas aplicaciones depende en gran medida de las garantías de QoS ofrecidas por la red.

En redes ad hoc, el soporte de la calidad de servicios se está convirtiendo en una necesidad inherente más que en una característica adicional de la red. A continuación explicamos las tres principales razones que hacen recomendable el diseño de redes ad hoc que ofrezcan QoS en vez de añadir algunas características como complemento.

- **El canal inalámbrico fluctúa rápidamente y estas fluctuaciones afectan severamente a los flujos multi-hop**. En oposición al Internet cableado, la capacidad de un canal inalámbrico fluctúa rápidamente debido a varios fenómenos que afectan a la capa física como la pérdida del enlace o las interferencias producidas por elementos del entorno. Además, el ruido de fondo y las interferencias de los nodos vecinos afectan a la calidad del canal inalámbrico. En redes ad hoc, la calidad de una conexión "extremo a extremo" puede variar rápidamente, ya que el cambio en la calidad de cualquier enlace puede afectar a la métrica de QoS "extremo a extremo" de los caminos multi-hop.

- **Competencia entre los paquetes por el medio compartido con los enlaces adyacentes a un flujo**. La contención entre paquetes del mismo flujo en diferentes nodos afecta intensamente a la calidad de una conexión. Tal contención surge de la compartición del canal inalámbrico por los nodos vecinos. A diferencia de Internet, este fenómeno afecta al QoS incluso en la ausencia de otro flujo de datos en la red.

- **Las interferencias pueden afectar a nodos más allá de los vecinos**. Los efectos de las interferencias se manifiestan en redes ad hoc cuando se emplea una sola frecuencia para comunicarse a través del medio compartido. En las redes inalámbricas con estructuras single-hop lo más normal es emplear un esquema de frecuencia cuando se pueden configurar las estaciones más cercanas para operar en distintas frecuencias y e reducir así las interferencias. Las transmisiones en el medio inalámbrico no se reciben correctamente más allá del rango de transmisión. Pero aún más allá del rango de transmisión, la energía restante puede ser suficiente para interferir con otras transmisiones. Por lo tanto,

interferencias de nodos no vecinos pueden resultar en la eliminación de paquetes.

Para soportar QoS en caminos multisalto, el protocolo debe estar diseñado para proporcionar QoS tanto entre el inicio y el final del camino como en cada salto. Las capas física y MAC son responsables de proporcionar QoS en saltos simples. La capa de red es responsable de la calidad de QoS en rutas completas.

El concepto de red ad hoc no depende de ninguna tecnología inalámbrica. Sin embargo, con la expansión de os dispositivos Wireless LAN (WLAN) en casa, oficinas y de las zonas wi-fi públicas, el término wirless se está convirtiendo en sinónimo de WLAN. Actualmente en las tiendas existen dos productos que se están convirtiendo en competidores de los standards WLAN, llamados IEEE 802.11a y IEEE 802.11b. Estos standards difieren del original 802.11 en la especificación de la capa física. Sin embargo, la capa MAC permanece inalterable en estos tres protocolos. De aquí en adelante, utilizaremos el término 802.11 para referirnos globalmente a los tres standards. La alta velocidad (sobre los 54 Mbps con 802.11a), la reducción de los precios y la proliferación de dispositivos portátiles con conexiones inalámbricas, son las tres razones principales de su popularidad.

La mayoría de los investigadores consideran a CSMA/CA basado en 802.11, la tecnología inalámbrica subyacente en redes ad hoc. Además se está investigando exhaustivamente la posibilidad de emplear otras técnicas de acceso al medio como TDMA, en redes ad hoc. Más recientemente, existe un creciente interés en aplicar las redes ad hoc en diferentes escenarios, como las redes ad hoc acústicas para la exploración marina. El standard 802.11 tiene dos modos de operación, llamados modo Infraestructura y modo ad hoc. Los modos se corresponden con las configuraciones WLAN y ad hoc respectivamente. En la configuración WLAN los nodos se comunican únicamente a través del punto de acceso (AP). En la configuración ad hoc, los nodos se comunican mediante saltos a través de enlaces inalámbricos punto a punto creados gracias a la proximidad con otros nodos. En la figura 2.2 podemos ver el el modo de funcionamiento de una red WLAN, mientras en la figura 2.1 podemos ver una red ad hoc formada por 7 equipos portátiles.

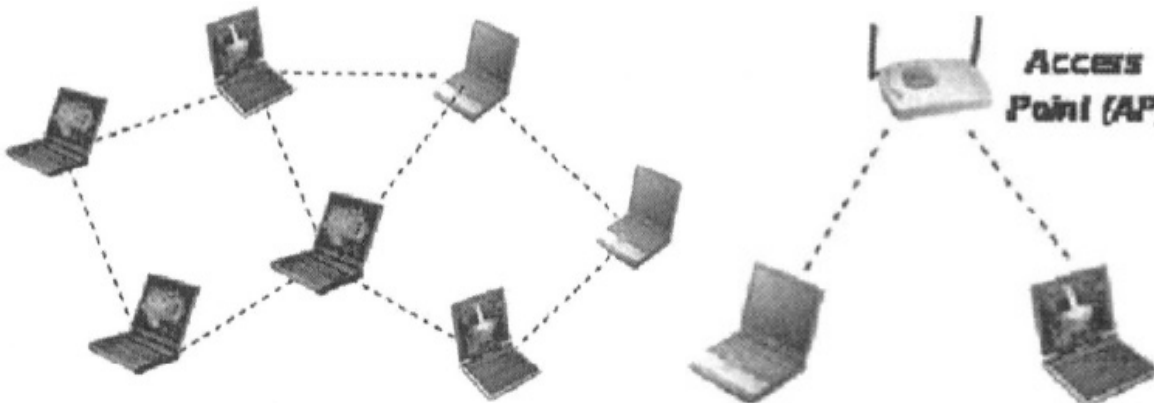


Fig .2.1. *Red ad hoc*

Fig 2.2. *WLAN*

2.1. Definición de QoS

El término calidad de servicio expresa diferentes conceptos en cada capa de la red. En la capa física, la QoS está relacionada con la velocidad de transmisión de datos y la tasa de pérdida de paquetes en los enlaces inalámbricos, lo cual depende de la calidad del canal. Si existe una variación continua de la calidad del canal, es importante mantener una tasa de envío de datos constante y una tasa de pérdida de paquetes baja. En la capa MAC, el QoS está relacionado con la fracción de tiempo durante la que cada nodo acceder al canal y transmitir un paquete satisfactoriamente. En la capa de red, las métricas "extremo a extremo" dependen de las métricas de cada salto realizado durante la ruta multihop. La capa de red debe calcular y mantener rutas que satisfagan los requisitos de QoS durante el tiempo que dure la conexión. La capa de transporte y sus superiores podrían incluir algun soporte para QoS si la capa de red no cumple los requisitos de QoS.

Ancho de banda, retardo y el jitter (variación de la tasa de envío) son las tres métricas de QoS más estudiadas. Sin embargo, el problema de la QoS en redes ad hoc presenta más desafíos que en una red cableada. Como resultado de los estudios realizados hasta ahora, se han aportado pequeños desarrollos para soportar retardo y jitter, pero el principal centro de atención de las investigaciones es proporcionar garantías de ancho de banda. Han sido propuestos varios mecanismos para estimar la cantidad de ancho de banda en redes CSMA/CA (Carrier Sense Multiple Access) y redes TDMA.

Para las redes ad hoc es muy difícil proporcionar garantías estrictas de QoS debido a las fluctuaciones del canal inalámbrico y a las interferencias de los nodos no vecinos. Es por ello más sencillo diseñar soluciones donde el soporte para QoS de la red es del tipo soft-assurances en vez de hard guarantees. Por el mismo motivo, son más comunes las garantías relativas a las garantías absolutas.

2.2. Señalización de QoS

La señalización de QoS es el proceso de establecimiento de una conexión desde un nodo fuente hasta un nodo destino que involucra la reserva de recursos en los nodos intermedios. La señalización de QoS actúa como un centro de control para el soporte de QoS. Reserva y libera recursos, establece, termina y renegocia flujos en las redes.
Los sistemas de señalización de QoS se pueden dividir en sistemas de señalización en-banda o fuera-de-banda. En la señalización en-banda, la información de control se viaja dentro de los mismos paquetes de datos (*piggybacking*), mientras que en la señalización fuera-de-banda, la información de control se envía en paquetes explícitos.

INSIGNIA es un ejemplo de sistema de señalización en-banda para el soporte de QoS en redes ad hoc. Cuenta con algoritmos rápidos de reserva de recursos, restablecimiento de rutas y adaptación por flujo, los cuales están específicamente diseñados para proporcionar un servicio adaptativo en tiempo real en un ambiente de redes ad hoc móviles. Para establecer un flujo adaptativo en tiempo real, la información de señalización se transporta en cada paquete IP de datos, en el campo que se conoce como opción INSIGNIA. Cuando un nodo intermedio recibe un paquete con el valor apropiado en el campo de opción INSIGNIA, reserva los recursos si están disponibles y

reenvía el paquete en dirección del nodo destino. El destino envía un mensaje de reporte de QoS a la fuente de forma periódica. El reporte de QoS indicará a la fuente el estado de la red. Este reporte puede tomar una ruta diferente hacia la fuente. La fuente toma decisiones de adaptación con base en el reporte de QoS. Todos los nodos intermedios mantienen información de estado del enlace (*soft state*). La ausencia de tráfico producirá la recuperación o liberación de los recursos asignados al flujo para que puedan ser utilizados por otros flujos. Otros mecanismos para el transporte de señales de QoS en redes ad hoc son el SWAN (*Service Differentiation in Stateless Wireless Ad hoc Networks*) [36] y el Courtesy Piggybacking.

2.3. Modelos de QoS

El modelo de QoS especifica la arquitectura en la cual ciertos servicios pueden ser proporcionados por la red. Un modelo de QoS para MANETs considerará primero las características de las redes, por ejemplo, la topología dinámica o la variación de la capacidad de los enlaces a lo largo del tiempo. Además las aplicaciones comerciales para MANETs requieren conexión a Internet. Por lo tanto, los modelos de QoS para MANETs consideran también ñas arquitecturas de QoS existentes para Internet. Vamos a comentar en primer lugar los modelos de QoS para Internet como IntServ y DivServ. A continuación, proponemos un nuevo modelo de QoS propuesto para MANETs.

2.3.1. Integrated Server (IntServ) y Resource Reservation Protocol (RSVP) en redes cableadas

La idea básica del modelo IntServ es que los estados de los flujos son mantenidos por cada router IntServ activo. Un flujo es una sesión de aplicación entre dos usuarios finales. Un estado específico de flujo incluirá los requisitos de ancho de banda, retardo asociado y coste del flujo. Además de su servicio Best Effort, IntServ propone dos clases de servicios, Guaranteed Service y Controlled Load Service.

El Guaranted Service se proporciona a aplicaciones que requieren un retardo fijo. El Controlled Load Service que requieren un servicio Best Effort fiable y mejorado. Debido a que la información del estado del flujo se mantiene en cada router. El QoS proporcionado por IntServ es para cada flujo individual. En un router con IntServ activo, IntServ se encuentra implementado con cuatro componentes fundamentales: el protocolo de señalización, la rutina de control de admisión, el clasificador y el planificador de paquetes. Otros componentes, como el agente de enrutamiento, son los originales del router y no es necesario cambiarlos. El RVSP se emplea como protocolo de señalización para reservar recursos en IntServ. Las aplicaciones con requisitos de Guaranteed Service o Controlled Load Service usan RSP para reservar recursos antes de la transmisión. El control de admisión decide aceptar o no la solicitud de recursos. Es invocado por cada nodo para decidir de forma local aceptar o rechazar una solicitud de un router de un servicio de tiempo real a través de algún camino de Internet. El control de admisión notifica a la aplicación a través de RSVP si el requerimiento de QoS puede ser satisfecho o no. La aplicación sólo puede transmitir sus paquetes de datos después de que la solicitud de QoS sea aceptada. Cuando un nodo recibe un paquete de datos, el clasificador realizará una clasificación multicampo (MF), la cual clasifica un paquete en función de varios campos como la dirección del origen y del destino, los números de puerto del origen y del destino, los

bits de tipo de servicio (TOS), y el ID del protocolo en la cabecera del Internet Protocol (IP). Después, el paquete clasificado, será colocado en la cola correspondiente de acuerdo con el resultado de la clasificación. Finalmente, el planificador de paquetes reordena la cola de salida en función de los diferentes requisitos de QoS.

El modelo IntServ-RVSP es no aplicable a MANETs debido a la limitación de los recursos en MANETs:

La cantidad de información de estado se incrementa proporcional con el número de flujos (el problema de escalabilidad, el cual es también un problema en el actual Internet). Mantener información sobre el estado de flujo supondrá un elevado coste de almacenamiento y sobrecarga de procesamiento para un terminal móvil, cuyas capacidades de almacenamiento y de procesamiento son limitadas. Aunque el problema de la escalabilidad no debería darse en las MANETs actuales debido a su ancho de banda limitado y su relativamente escaso número de flujos comparado con redes cableadas, esto dejará de ocurrir con el desarrollo de tecnologías de radio más rápidas y el elevado número de usuarios potenciales en un futuro cercano.
Los paquetes de señalización de RSVP competirán por el uso del ancho de banda con los paquetes de datos y consumirán un porcentaje sustancial de ancho de banda en MANETs.
Cada terminal móvil debe realizar el procesamiento del control de admisión, clasificación y planificación. Esto es una gran carga para los limitados recursos de los terminales móviles.

2.3.2. Differentiated Service (DiffServ)

El Differentiated Service (DiffServ) está diseñado para superar las dificultades de implementar y utilizar IntServ y RSVP en Internet. DiffServ proporciona un número limitado de clases agregadas para resolver el problema de la estabilidad de IntServ. Diffserv define la distribución de los bits TOS in la cabecera IP, llamado campo DS y un conjunto básico de reglas de envío de paquetes llamadas “Per-Hop Behavior” (PHB). En los límites de la red, los routers periféricos controlan el tráfico entrante en la red con mecanismos de clasificación, marcado y política. Cuando un pquete de datos entra en un dominio con DiffServ, un router periférico marca el campo DS del paquete, y los routers interiores reenvían el paquete a lo largo del camino de envío basándose en el campo DS. Debido a que el campo DS sólo codifica un conjunto limitado de clases de servicios, el cálculo de los nodos interiores es muy sencillo y rápido. A diferencia de IntServ, los routers interiores en DiffServ no necesitan mantener información de estado por flujo.

Muchos servicios, como El Premium Service, Assured Service y Olympic Service, pueden ser soportados por el modelo DiffServ. Premium Service debe proporcionar bajas pérdidas, bajo retardo, bajo Ritter y garantías de ancho de banda “extremo a extremo”. Assured Service es para aplicaciones que requieren una mayor fiabilidad que la proporcionada por el Best Effort Service. Su propósito es proporcionar una productividad garantizada o al menos predecible para las aplicaciones. Además, es más cualitativa que cuantitativa, con lo que es más fácil de implementar. Olympic Service proporciona tres grados de servicio (Oro, plata y bronce) ºcon calidad decreciente. Soportar Premium Service es casi imposible en redes MANETs.

Diffserv podría ser una solución como modelo de QoS para MANETs debido a su ligereza en routers interiores. Además, proporcionar Assured Service, el cual es un servicio viable en el contexto de las MANETs. Sin embargo, debido a que DiffServ está diseñado para redes cableadas fijas, por lo que habría que modificar algunas características para implementar DiffServ en MANETs. Primero, es ambiguo definir los routers periféricos en MANETs. Intuitivamente, los nodos emisores desempeñarían el papel de nodos periféricos. Otros nodos a lo largo de la ruta de envío del origen al destino serán los nodos interiores. Pero todos los nodos deben tener la funcionalidad de un nodo interior y de un nodo periférico, puesto que las fuentes no están predefinidas. Esto provocará un alto coste de almacenamiento en cada terminal. Segundo, el concepto de un Service Level Agreement (SLA) en Internet, no está definido en MANETs. El SLA es una especie de contacto entre un cliente y un Internet Service Provider (ISP) que especifica los servicios de envío que el cliente recibe. En Internet, un cliente debe tener un SLA con su ISP para recibir Differentiated Services. El SLA es indispensable porque incluye todas o parte de las reglas que condicionarán el tráfico. Los reguladores del tráfico son situados en los nodos donde se origina el tráfico; son los encargados de remarcar los flujos de tráfico, descartar o configurar los paquetes de acuerdo con el perfil de tráfico, que describe las propiedades temporales de un flujo. Implementar un SLA in MANETs es difícil porque no hay un esquema trivial para que los nodos móviles negocien sus reglas de tráfico.

2.3.3. Flexible QoS Model for Mobile Ad Hoc Network (FQMM)

FQMM consider alas características de las MANETs e intenta sacar provecho de ambos modelos anteriormente expuestos. La separación de servicio por flujo de IntServ y la diferenciación de servicios de DiffServ. Como en DiffServ, tres tipos de nodos (ingress, interior, y egress) son definidos en FQMM. Un nodo ingress en un nodo móvil que envía datos. Nodos interiores son nodos que reenvían datos de otros nodos. Un nodo egress es un nodo destino. El papel de un nodo móvil es adaptable dependiendo la posición que ocupe en la red. La provisión de servicios en FQMM, el cual es empleado para determinar y asignar recursos a varios nodos, es un modelo híbrido de la provisión por flujo como en IntServ y la provisión por clase como en DiffServ. FQMM intenta preservar la granularidad por flujo para una pequeña parte del tráfico de la MANET, dado que una gran cantidad del tráfico podrá ser agregada al flujo, es decir, granularidad por clase.

FQMM es el primer intento de proporcionar un modelo de QoS para MANETs. Sin embargo, algunos problemas aún deben ser resueltos. Primero, cuantas sesiones para diferentes flujos pueden ser mantenidas. Sin un control específico del número de servicios con granularidad por flujo, el problema de la escalabilidad todavía existe. Segundo, como en DiffServ, los nodos interiores envían paquetes de acuerdo a un cierto PHB que es escrito en el campo DS.

2.4. QoS en las diferentes capas

2.4.1. Capa física

Uno d los desafíos fundamentales en redes inalámbricas es el continuo cambio de las propiedades físicas del canal. Las capas físicas de 802.11a y 802.11b pueden soportar diferentes tasas de envío. Dependiendo de la calidad del canal la tasa de envío puede modificarse para mantener una tasa de errores aceptable, puesto que las tasas de envío elevadas son susceptibles de presentar altas tasas de bits erróneos.

El standard 802.11a opera en la banda de los 5.7 GHz y soporta tasa de envíos de 6, 9, 12, 18, 24, 36,48 y 54 Mbps. El standard 802.11b opera en la banda de los 2.4 GHz y soporta 1, 2, 5.5 y 11 Mbps. Sin embargo, estos standards no proporcionan ninguna manera de detectar la tasa máxima de un enlace.

La política de selección de la tasa de envíos tiene un impacto directo en las métricas de QoS del canal. Por ejemplo, la política de selección más conservadora consiste en escoger siempre la tasa más baja en todos los enlaces de la red ad hoc. Si una aplicación requiere que todos los enlaces tengan la misma tasa de envíos, la política de la menor tasa de envíos puede funcionar. Sin embargo, provoca una importante infrautilización de los recursos, ya que los enlaces con buena calidad de canal no envían a su máxima velocidad.

Para un uso eficiente de una capa física con tasas de envíos heterogéneas, existen varios algoritmos propuestos. Algunos de estos algoritmos están fuertemente ligados a la capa MAC. Su funcionamiento repercute en la productividad observada en un enlace y la productividad "extremo a extremo" en las conexiones multi-hop. Los requisitos de QoS de las capas superiores pueden afectar al diseño de este algoritmo. Sin embargo, los propósitos actuales se centran en optimizar la utilización de los enlaces, aunque pueden ser modificados para implementar los requisitos de QoS de capas superiores.

2.4.1.1. Auto Rate Fallback (ARF)

El Auto Rate Fallback (ARF) es un algoritmo que intenta encontrar la mayor tasa de envío posible en un enlace. Fue diseñado para los dispositivos Lucent's WaveIan II basados en el standard IEEE 802.11b. Por defecto, operan con la mayor tasa de envío posible. Cuando un ACK de la capa MAC se pierde tras una transmisión correcta de datos, la primera retransmisión se realiza con la misma tasa. Si se vuelve a perder el ACK, la tasa se reduce a la siguiente tasa de envíos en las transmisiones y retransmisiones posteriores. Si se reciben diez ACKs correctamente o si un temporizador finaliza su tiempo de espera, el dispositivo intenta actualizar la tasa de envío. Si la primera transmisión con una tasa mayor falla, se vuelve inmediatamente a la tasa de envío anterior.

2.4.1.2. Receiver-Based Auto Rate (RBAR)

Este protocolo conocido como RBAR (Receiver-Based Auto Rate) opera en la capa MAC y puede adaptarse a las fluctuaciones del canal. G. Holland, Nitin Vaidya, and Paramvir Bahl, observaron que la tasa de envío de un enlace 802.11 puede fluctuar con mucha frecuencia (alrededor de 50 veces por segundo) y el algoritmo ARF no es capaz de adaptar su tasa de envío de acuerdo con las condiciones del canal. El algoritmo emplea un intercambio de paquetes RTS-CTS en modo 802.11 DCF para obtener más

información de las condiciones del canal. El SNR (Signal to Noise Ratio) del RTS se emplea para determinar la mayor tasa de envíos posible que puede ser usada para transmitir paquetes de datos. La máxima tasa de envío permitida es notificada al emisor usando el CTS. Ya que la estimación de la tasa de envío es realizada por el receptor inmediatamente antes del comienzo de la transmisión de datos, la estimación es muy exacta.

2.4.1.3. Opportunistic Auto Rate (OAR)

B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly proponent un mecanismo llamado Oppotunistic Auto Rate (OAR) para la mejora de la productividad en presencia de enlaces con diferentes tasas en redes ad hoc. La idea principal es enviar múltiples paquetes cuando la tasa es máxima. El protocolo RBAR puede emplearse para calcular la tasa que puede ser soportada por el canal. De forma similar, OAR puede ser empleado con protocolos adaptativos basados en la tasa del emisor como ARF. Sin embargo, está demostrado que RBAR mejora las prestaciones de ARF. El algoritmo garantiza que se concede a todos los nodos acceso al medio durante el mismo tiempo permitido por IEE802.11 con enlaces simples. Este mecanismo oportunista es similar al diseño de un algoritmo de planificación proporcionalmente justo para redes 3G como HDR (High Data Rate).

2.4.2. Capa de Acceso al Medio

El standard original de IEEE 802.11 especifica que las capa física y MAC proporcionan una tasa de envío de 2 Mbps. Los standard posteriores IEEE 802.11b y IEEE 802.11ª modifican la capa física y aumentan su tasa máxima hasta 11 Mbps y 54 Mbps respectivamente.

A continuación vamos a comentar el funcionamiento básico de la capa MAC en 802.11, que emplea el protocolo Distributed Coordination Function (DCF) para distribuir el acceso al medio compartido. También comentaremos el Point Coordination Function (PCF) el cual proporciona un mecanismo centralizado de acceso al medio. DCF es la elección natural cuando se trabaja con redes ad hoc puesto que no necesita un controlador central. Sin embargo, PCF puede soportar métricas de QoS en redes inalámbricas single-hop debido a su diseño centralizado. Ambos están presentes en el nuevo standard 802.11e, el cual está diseñado para soportar QoS en WLANs. Posteriormente presentamos alunas características importantes del protocolo 802.11e y algunos esquemas de diferenciación de servicios que han sido propuestos como extensión del DCF.

2.4.2.1. 802.11 Distributed Coordination Function (DCF)

El protocolo DCF se encarga de proporcionar igual acceso (en términos de número de paquetes) a todos los nodos en espera que comparten el canal. Por ejemplo, en modo Infraestructura si todos los nodos en una región están dentro del rango de transmisión del resto y no hay otras fuentes de ruido o de interferencias, todos los nodos y el AP pueden enviar el mismo número de paquetes.

En redes ad hoc la productividad de cada nodo usando DCF es una función del número de vecinos que tiene y del estado de sus colas. Ya que la productividad de los vecinos depende a su vez de sus vecinos, la productividad pasa a ser un problema global en vez de un problema local. Por lo tanto, en general en una red ad hoc usando DCF la productividad recibida por un nodo depende de la topología completa. Hay que tener en cuenta que el mecanismo DCF se encarga de proporcionar acceso por nodo y no por enlace.

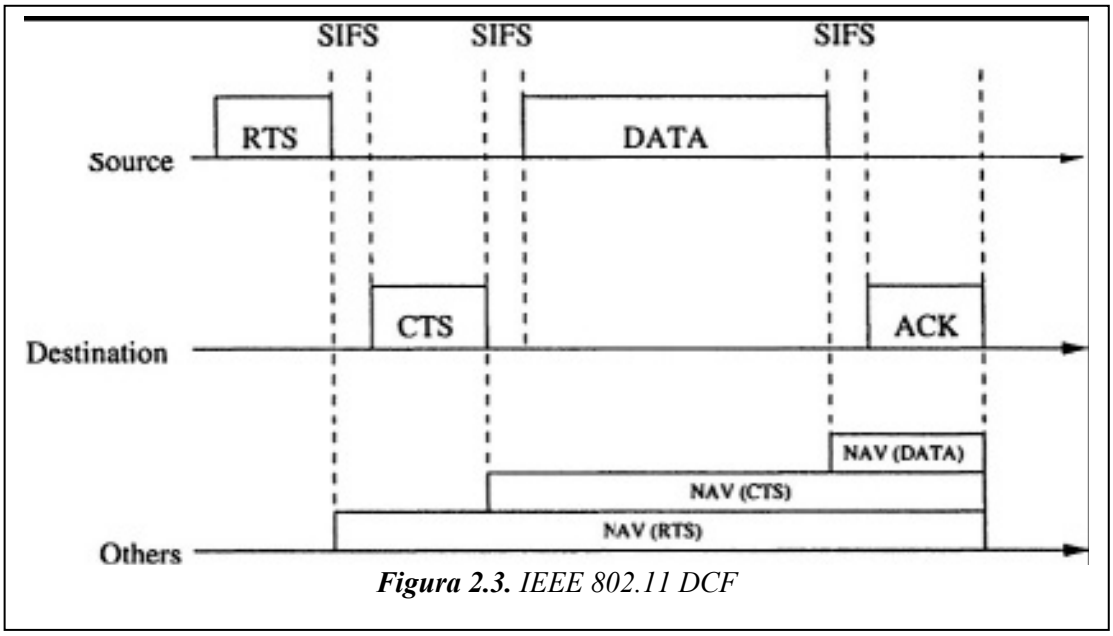


Figura 2.3. *IEEE 802.11 DCF*

Describamos el funcionamiento de DCF en detalle. A cada nodo que tiene un paquete para enviar se le asigna una ranura para transmitir en [0, *cw*], donde cw es la ventana de contención usada para el backoff. Inicialmente se asigna a *cw* el valor cw_{min}. En la ranura asignada, el nodo envía un paquete de control de capa MAC llamado RTS (request-to-send), al receptor. Si el receptor recibe correctamente el RTS y no retrasa la transmisión, responde con un CTS (clear-to-send). A continuación, el emisor envía los paquetes de datos que son recibidos por el receptor. Las transmisiones de estos cuatro paquetes están separadas por un pequeño intervalo de tiempo conocido como SIFS (Short Inter-Frame Space). El SIFS permite al nodo tener tiempo de cambiar al transmisor entre el modo emisor y receptor. La secuencia de transmisión de estos cuatro paquetes se muestra en la Figura 2.3. La cabecera MAC de todos estos paquetes (ver la estructura de los paquetes en la Figura 2.4) contiene un campo "duración" indicando el tiempo restante hasta la recepción del paquete ACK. Basándose en este pronóstico, los nodos vecinos actualizan una estructura de datos llamada NAV (Network Allocation Vector). Esta estructura mantiene el tiempo restante durante el que cada nodo tiene que aplazar sus transmisiones.

Si la transmisión del paquete falla. El emisor dobla su ventana de contención (*cw* ← [2 x *cw* - 1]) y lo pone en espera antes de intentar reenviarlo. El número de retransmisiones está limitado a cuatro para paquetes pequeños (incluidos paquetes RTS) y siete para paquetes largos (generalmente paquetes de datos). Si se supera estos contadores, el paquete de datos es eliminado y *cw* es reiniciado a cw_{min}. Si el paquete de datos es enviado satisfactoriamente, tanto el emisor como el receptor reinician su valor *cw* a cw_{min}.

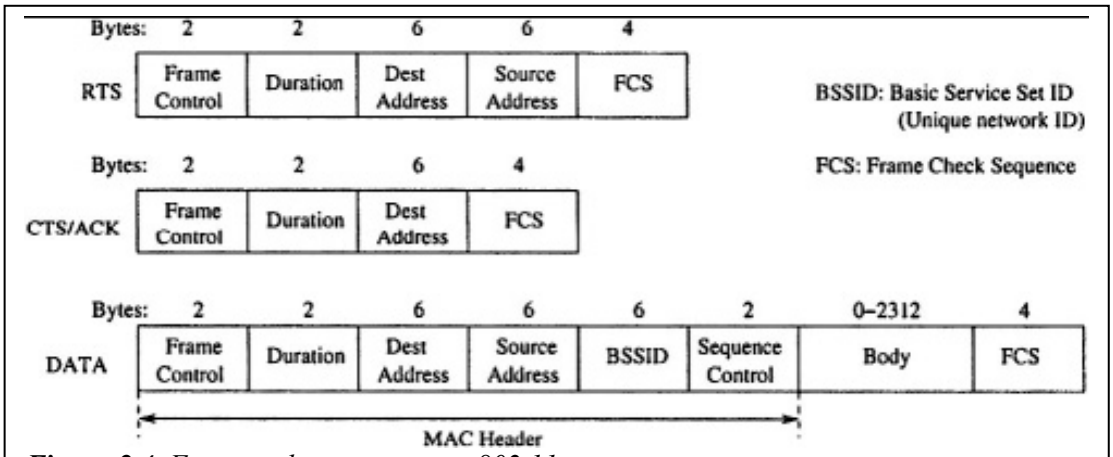


Figura 2.4. *Formato de paquete para 802.11*

2.4.2.2. 802.11 Point Coordination Function (PCF)

PCF opera en el modo Infraestructura de 802.11. El standard require que un AP implemente el modo PCF (período contención - libertad) y debe alternarlo con el modo DCF (período de contención). En el modo PCF, el point-coordinator (AP) envía paquetes a otros nodos y pregunta a una lista de nodos, dándoles la oportunidad de transmitir. A diferencia del modo DFC, en el modo PCF los nodos sólo pueden transmitir si son preguntados por el AP. El comienzo del período contención-libertad (el período en que opera PCF) está marcado por una señal del AP, el cual avisa también la duración del período. Durante este período, la planificación de las transmisiones está completamente controlada por el AP. La duración de este período puede ser reducida por el AP transmitiendo un paquete especial llamado CF-End packet. La solicitud y el reconocimiento son piggybacked en la paquetes de datos como se muestra en la Figura 2.5. Nótese que antes de enviar la señal, el AP espera durante un intervalo de tiempo llamado PIFS (PCF Inter Frame Space) el cual es más largo que el SIFS. Esto asegura que todas las transmisiones relacionadas con el período de contención hayan cesado. El intervalo PIFS se usa también para esperar una respuesta a la encuesta realizada por el AP. Una vez finaliza este intervalo, el AP concluye que el nodo encuestado no tiene ningún paquete para enviar o no ha recibido la pregunta. El continua preguntando al siguiente nodo.

2.4.2.3. La extensión QoS: 802.11e

La extensión 802.11e proporciona mecanismos para soportar diferentes prioridades en redes WLAN. Siendo un protocolo distribuido, es difícil cumplir prioridades estrictas. De ahí que las prioridades sean de naturaleza probabilística. Por tanto, las prioridades pueden ser vistas como una forma de métrica de QoS.

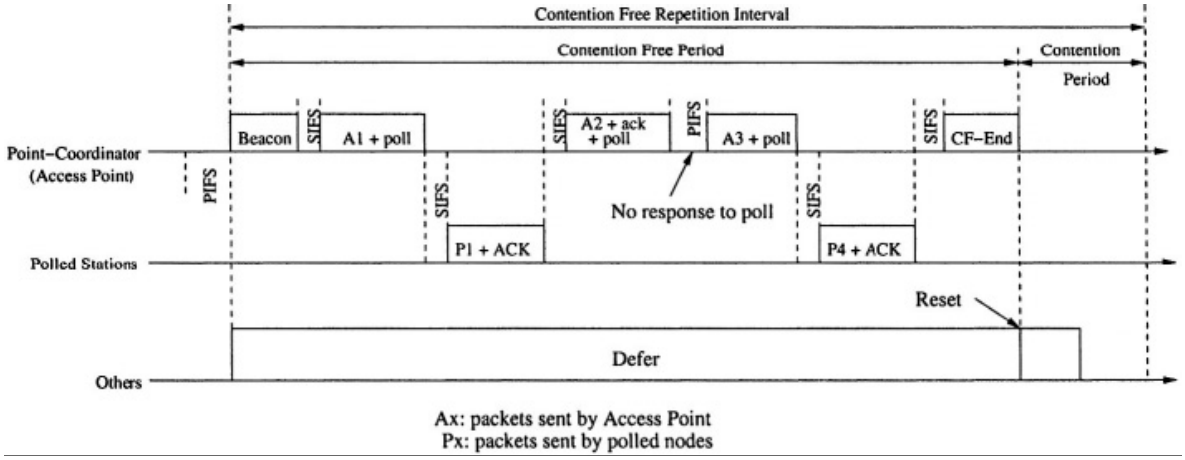


Figura 2.5. *Point Coordination Function.*

Las funcionalidades de DCF y PCF de 802.11 han sido extendidas, y estas funcionalidades constituyen el standard 802.11e (el proceso de standarización no está complicado). El Enhanced DCF (EDCF) extiende la funcionalidad de DCF para proporcionar el concepto de prioridad. La evolución de PCF es conocida como HCF (Hybrid Coordination Function) en 802.11e. Algunos de los mecanismos de 802.11e son similares a los servicios de diferenciación sobre los que hablaremos más adelante. La figura 2.6 muestra el funcionamiento del 802.11e detalladamente.

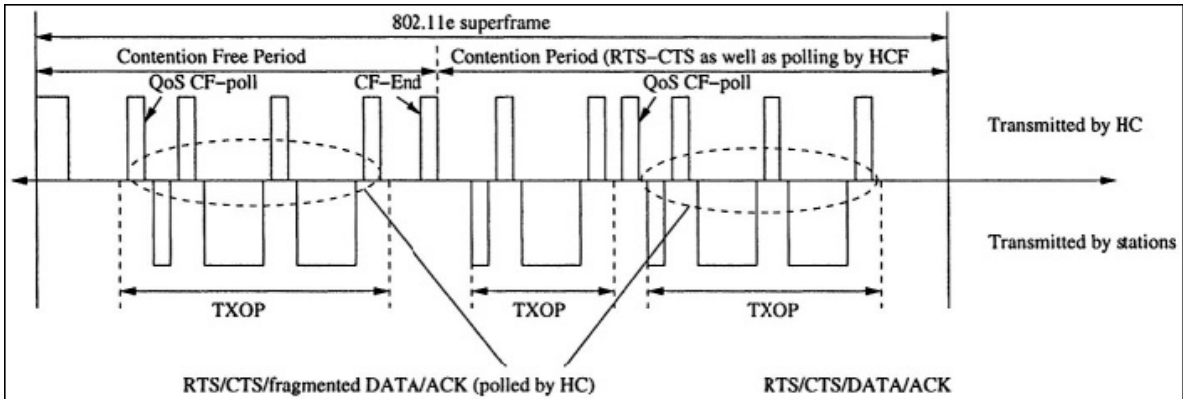


Figura 2.6. *Ejemplo de un super-frame 802.11*

En EDCF, los elementos que accedan a la capa MAC pueden solicitor ocho prioridades de servicio diferentes. Estas prioridades son traducidas a diferentes categorías de acceso (ACs). Cada C puede tener un valor distinto para el período DIFS (llamado aquí AIFS), cw_{min} y cw_{max}. La Figura 2.7 muestra un ejemplo dibujando diferentes clases de tráfico con diferentes valores de AIFS. Estos valores pueden ser determinados dinámicamente por el AP. Los nodos son informados de estos nuevos valores a través de señalizadores. Los diferente niveles de prioridad se corresponden con diferentes valores de AIFS.

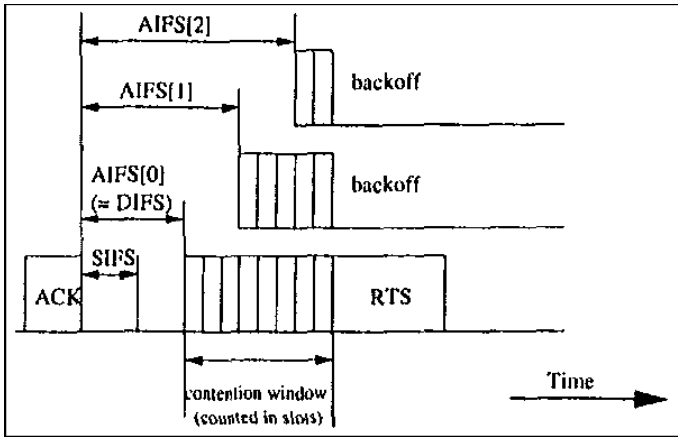


Figura 2.7. *Backoff múltiple de streams con diferentes prioridades.*

HCF permite al coordinador híbrido mantener el estado de los nodos y permite repartir los turnos de transmisión (TXOP) de forma inteligente. A diferencia del modo PCF del 802.11, el coordinador híbrido puede pregunta a los nodos en el período de contención – libertad así como en el período de contención.

Al igual que PCF en 802.11, este protocolo requiere un modo de operación centralizado. Para satisfacer los requisitos de QoS, el AP coordina las transmisiones en su zona. Este protocolo debe ser extendido para redes ad hoc donde no hay un coordinador central.

2.4.2.4. Soporte para QoS usando DCF basado en diferenciación de servicios

Así como es difícil proporcionar garantías absolutas de QoS, proporcionar garantías relativas de QoS es posible mediante la diferenciación de servicios. Esto ayuda en el diseño de sistemas que pueden soportar diferentes clases de usuarios.

Como comentamos anteriormente, en 802.11 todos los nodos en espera compitiendo por el canal usan el mismo protocolo con el mismo conjunto de parámetros. Como resultado, si todos los nodos competidores se encuentran cada uno en el rango de los otros, 802.11 proporcionará a largo plazo un tiempo similar a cada nodo. Sin embargo, para proporcionar diferentes servicios, el protocolo 802.11 debe ser modificado. G. Armitage propone tres formas de modificar la funcionalidad de 802.11 para soportar diferenciación de servicios. Los parámetros que es necesario modificar para obtener una diferenciación de servicio son los siguientes:

- *Backoff increase function*: tras un intento fallido de enviar un paquete RTS o de datos, el valor máximo de backoff se dobla. Concretamente el nuevo valor se calcula de la siguiente manera:

  $$Backoff_{time} = |\_2^{(2+i)} \text{ x } rand()\_| \text{ x } Slot_{time}$$

  donde *i* es el número de backoffs consecutivas que ha experimentado el paquete para ser transmitido. Para soportar diferentes prioridades, el cálculo del backoff puede ser modificado como sigue:

$$Backoff_{time} = \lfloor Pj^{(2+i)} \text{ x } rand() \rfloor \text{ x } Slot_{time}$$

donde *Pj* es la prioridad del nodo *j*.

- *DIFS*: como muestra la Figura 2.2, este el mínimo intervalo de tiempo requerido antes de iniciar la transmisión de un nuevo paquete después de que el canal haya estado ocupado. Para reducir la prioridad de un flujo podemos incrementar el período DIFS de los paquetes de dicho flujo. Sin embargo, es difícil encontrar una relación directa entre el período DIFS de un flujo y su productividad. La figura 2.8 muestra los valores diferentes del DIFS y las prioridades relativas correspondientes. Esta idea es similar al concepto de AIFS en 802.11e descrito en la sección anterior.

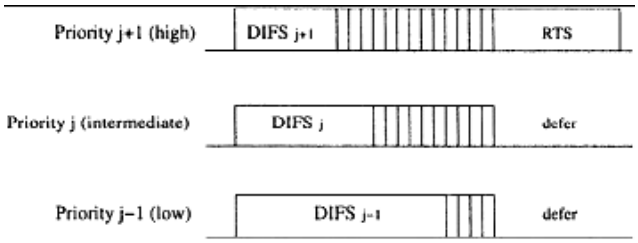


Figura 2.8. *Diferencicación de servicio usando el valor de DIFS.*

- *Maximum Frame Length*: La contención del canal usando la funcionalidad DCF es usada normalmente para enviar un mensaje sencillo. Para enviar mensajes más largos, se pueden obtener un mayor throughput con flujos con prioridad mayor.

2.4.3. Enrutamiento con QoS

Las métricas de QoS de un recorrido "extremo a extremo" depende de los enlaces que forman la ruta. Existen tres dificultades principales para calcular una ruta que satisfaga los requisitos de QoS. Primero, la métrica de QoS de cada enlace debe ser calculada de forma constante o bajo demanda, cuando el paquete *route request* es enviado.Segundo, el broadcast basado en algoritmos de enrutamiento no exploran todas las rutas posibles. Tercero, los mecanismos para calcular el ancho de banda de un enlace tienen limitaciones ya que se basan en observar otros parámetros como la longitud de la cola y la historia de acceso al canal.

Las redes multi-hop son dinámicas por naturaleza, y las conexiones son susceptibles de perderse, sufrir interferencias y colisiones ocasionadas por terminales ocultos o expuestos. Estas características hacen que diseñar un algoritmo de enrutamiento con QoS para redes multi-hop sea un desafío. Estos serán los objetivos que deberá alcanzar el algoritmo:

- El algoritmo deberá ser muy robusto y no degradarse en exceso con el aumento de la movilidad.
- El cálculo de rutas no debe requerir el conocimiento de información global.
- La ruta calculada debe mantener el ancho de banda solicitado por el flujo.
- El cálculo de la ruta debe involucrar al menor número de nodos posibles, puesto que la inundación completa de la red es muy cara.

- Los hosts deben tener un acceso muy rápido a las rutas cuando se desea establecer una conexión.

AODV (Ad hoc On-demand Distance Vector) y DSR (Dynamic Source Routing) fueron de los primeros protocolos de enrutamiento propuestos para redes ad hoc. Ambos protocolos son reactivos (bajo demanda). AODV emplea enrutamiento *hop-by-hop,* es decir, cada nodo intermedio indica al paquete el siguiente salto que debe tomar para alcanzar el destino. Por contra, el DSR emplea enrutamiento en el origen, es decir, el emisor indica al paquete toda la ruta que debe seguir hasta su destino. Existen extensiones propuestas para AODV en redes TDMA y para DSR que soportan QoS

Algunos protocolos han sido diseñados para soportar QoS, en lugar de intentar incluir posteriormente QoS al protocolo. Vamos a describir a continuación dos ejemplos.

2.4.3.1. Core Extraction based Distributed Ad hoc Routing (CEDAR)

CEDAR permite superar algunos retos de diseño para redes ad hoc de pequeño o medio tamaño, redes que consistan en algunos cientos de nodos. CEDAR tiene tres componentes principales:

- **Core Extraction**: un conjunto de hosts es distribuido y elegido dinámicamente para formar el núcleo de la red que constituya un conjunto representativo de la red usando únicamente cálculos locales y el estado local. La Figura 8.9 muestra un ejemplo de red con un núcleo de cuatro nodos. Cada nodo mantiene la topología local de los nodos de su domino, y también realiza el cálculo de rutas en lugar de estos nodos.

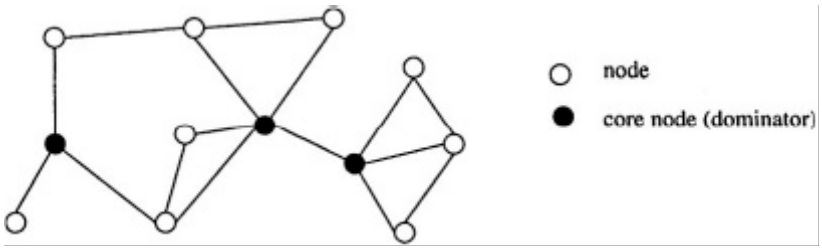


Figura 2.9. *CEDAR.*

- **Link State propagation**: CEDAR logra proporcionar QoS propagando la información sobre el ancho de banda disponible en los enlaces estables del grafo del núcleo. La idea básica es que la información sobre los enlaces estables con máximo ancho de banda pueda darse a conocer a nodos lejanos en la red, mientras que la información sobre enlaces dinámicos o con bajo ancho de banda permanece local.

- **Route computation**: el cálculo de rutas establece primero una ruta principal entre su nodo "jefe" y el de su destino. Este camino principal proporciona la dirección de la ruta del origen al destino. Usando esta información direccional, CEDAR calcula una ruta adyacente a la principal que satisfaga los requisitos de QoS.

2.4.3.2. Ticket based Routing

Ticket based Routing se basa en la idea de limitar los mensajes broadcast y direccionarlos a través de su dirección correcta. El objetivo de este enfoque es seleccionar rutas. La fuente tiene un cierto número de tickets. Los tickets son de dos tipos: amarillos y verdes. Cada exploración lleva un cierto número de tickets. El cometido de los tickets amarillos es maximizar la probabilidad de encontrar un camino viable. Por lo tanto, los sondeos llevando tickets amarillos prefieren caminos con retardos más bajos. El propósito de los tickets verdes es maximizar la probabilidad de encontrar un camino de bajo coste, donde cada enlace tiene asociado un coste. Los tickets verdes prefieren caminos con costes más pequeños, los cuales, sin embargo, podrían tener un retardo mayor, y por tanto, menos posibilidades de satisfacer los requisitos de retardo.

El emisor comienza el sondeo con un cierto número de tickets de cada color. Cada nodo intermedio toma la decisión de cuantos tickets serán reenviados en cada uno de los nuevos sondeos. Esta decisión se basa en las métricas de QoS que se conocen de los enlaces. Por ejemplo, un enlace con un retardo pequeño, tiene un mayor número de tickets amarillos comparado con un enlace con mayor retardo.

El Enhanced Ticket Based Routing Algorithm elimina los sondeos redundantes y aumenta la optimización del sondeo de tickets.

2.4.4. QoS en otras capas de la red

La necesidad de QoS procedee de la capa de aplicación. La capa de aplicación solicita a la capa de transporte la provisión de ciertos servicios. La capa de transporte debe consultar a la capa de encaminamiento para calcular rutas que satisfagan los requisitos del QoS. Estas consultas puede que deban recorrer la capa física. Cada capa que recibe una consulta QoS de la capa superior debe hacer lo siguiente:

- Comprobar si tiene soporte: Cada capa debe ver si los requisitos de QoS están entre sus límites. Debe notificárselo a la capa superior en caso de no ser capaz.

- Consultar a la capa inferior para ver si lo soporta: La capa que se encuentra actualmente procesando la consulta QoS debe ser capaz de darle soporte con la ayuda de las capas inferiores. Debe ser capaz de mapear los requisitos de los servicios que deben prestar las capas inferiores y enviar la consulta a dichas capas. Por ejemplo, el soporte de encaminamiento a una ruta QoS con un cierto mínimo de ancho de banda, la capa de encaminamiento debe informar a la capa MAC para que incremente la prioridad del canal de acceso.

- Negociar con las capas superiores/inferiores: cuando una consulta QoS es recibida desde una capa superior, se debe comprobar si la red soporta esa consulta. Si las necesidades de QoS no pueden ser conseguidas, un requisito QoS diferente puede ser negociado mediante el empleo de valores alternativos para las mediciones relevantes del QoS.

- Informar a la capa de aplicación sobre fallos en mantener la QoS: después de establecer una conexión QoS, en caso que la red falle al mantener las medidas del QoS, la capa de aplicación necesita ser informada para tomar las medidas apropiadas. Por ejemplo, si la red no es capaz de encontrar rutas que requieran un mínimo de ancho de banda para dar soporte a comunicación real, la capa de aplicación puede cambiar la codificación o resolución de los datos multimedia. La capa de red dándose cuenta de las modificaciones en la QoS debe informar a las capas superiores hasta llegar a la capa de aplicación.

2.5. Esquemas de QoS

Las soluciones para proporcionar QoS se clasifican normalmente en esquemas de QoS y las capas en las que operan. Anteriormente hemos realizado un análisis de las soluciones que existen para proporcionar QoS en cada capa. En esta sección analizaremos sistemas completos que proporcionan QoS. Estos esquemas pueden clasificarse en los siguientes tipos.

- **Ligados**: Existe una fuerte dependencia entre el algoritmo de enrutamiento y el mecanismo de QoS que proporciona las garantías de QoS. Ejemplos: Ticket-Based Probing (TBP), Predictive Location-Based QoS Routing Protocol (PLBQR), Time Domain Reflectometry (TDR), Quality of Service Ad Hoc On-Demand Distance Vector (QoS-AODV), BR, OQR, OLMOR, Active Query Router (AQR), Core-Extraction Distributed Ad Hoc Routing (CEDAR), y Intelligent Optimization Self-Regulated Adjustment (INORA).

- **Desligados**: no existe dependencias entre el protocolo de enrutamiento y el mecanismo de QoS. Ejemplos: INSIGNIA, stateless wireless ad hoc network (SWAN), y PRTMAC.

- **Independientes**: los protocolos de la capa de red no dependen de la implementación de la capa MAC. Ejemplos: TBP, PLBQR, QoS-AODV, INSIGNIA, INORA, y SWAN.

- **Dependientes**: los protocolos de red dependen de la capa MAC. Ejemplos: TDR, BR, OQR, OLMQR, AQR, CEDAR, and PRTMAC.

- **Table driven**: Será la tabla de rutas de cada nodo quien ayude a la transmisión de los paquetes. Ejemplo: PLBQR.

- **On demand**: El nodo emisor será el que encuentre la ruta. Ejemplos: TBP, TDR, QoS-AODV, OQR, OLMQR, AQR, INORA, y PRTMAC.

- **Híbridas**: Incluyen características de los esquemas on-demand y table driven. Ejemplos: BR and CEDAR

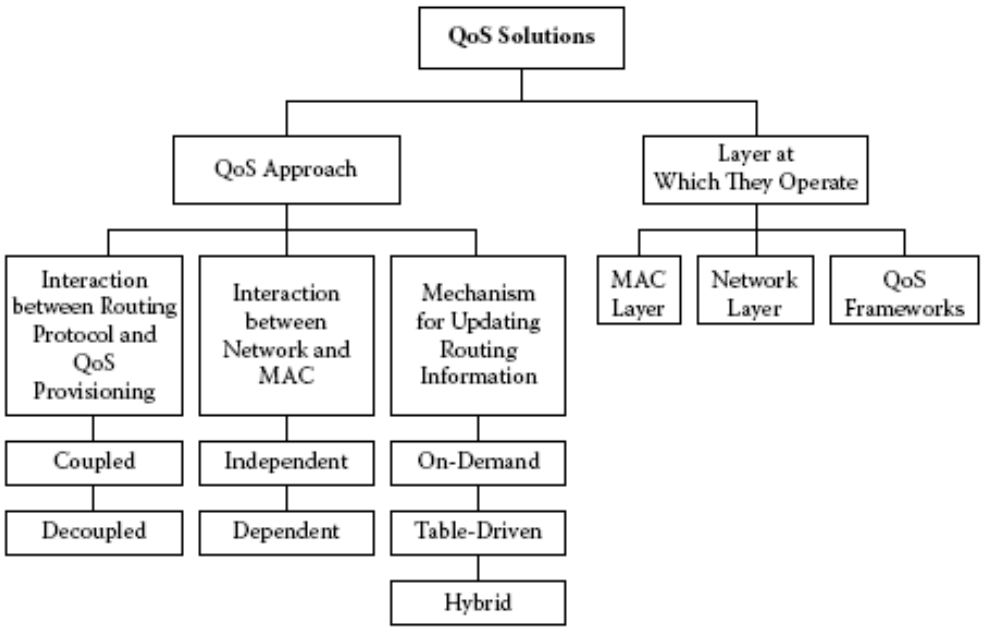


Figura 2.10.*Clasificación de las soluciones QoS.*

2.5.1. INSIGNIA

El esquema INSIGNIA permite a las aplicaciones empaquetadas de audio, video y de transmisión de datos en tiempo real especificar sus necesidades máximas y mínimas de ancho de banda, y juega un papel central en la asignación de recursos, restablecimiento del control, y la adaptación de las sesiones entre nodos móviles que se comunican. Basado en la disponibilidad de ancho de banda "extremo a extremo", los mecanismos de QoS intentan proporcionar garantía soportando servicios adaptativos. Para mantener servicios adaptativos, el esquema de QoS INSIGNIA comprende los siguientes elementos arquitectónicos, como mostramos en la Figura :

- Señalización en banda, que establece, restaura, adapta y desconecta servicios adaptativos entre pares origen-destino. Los algoritmos para restaurar el flujo responden a cambios dinámicos en el enrutamiento, y los algoritmos de adaptación responden a cambios en la disponibilidad del ancho de banda. Basado en una aproximación en las señales en línea que explícitamente lleva información de controlen la cabecera del paquete IP, los flujos y las sesiones pueden ser rápidamente establecidas, restauradas y publicadas en respuesta a alteraciones y cambios en la topología de la red inalámbrica.

- Control de admisión, que es responsable de asignar el ancho de banda a los flujos basándose en el máximo y mínimo ancho de banda solicitado. Una vez que los recursos han sido asignados, son periódicamente refrescados por un mecanismo a través de la recepción de paquetes de datos. El testeo del control de admisión está basado en medidas de la capacidad del canal y de su utilización y el ancho de banda requerido. Para mantener un protocolo de señalización simple y ligero, las nuevas reservas no afectan a las ya existentes.

- El envío de paquetes, que clasifica los paquetes entrantes y los envía al módulo apropiado (enrutamiento, señalización, aplicaciones locales. Modulos de

planificación de paquetes). Los mensajes de señalización son procesados para extraer la señalización en banda, y los paquetes de datos son entregados localmente o enviados al módulo de planificación de paquetes para transmitirlos al siguiente salto.

- El protocolo de enrutamiento, que rastrea dinámicamente los cambios en la topología de la redad hoc, creando una tabla de rutas visible para los mecanismos de envío de paquetes de cada nodo. El esquema QoS asume la disponibilidad de un conjunto genérico de protocolos de enrutamiento para MANETs que pueden ser conectados a la arquitectura. El esquema QoS asume que el protocolo de enrutamiento proporciona nuevas rutas, bien de manera períodica o por demanda, en caso de que se produzcan cambios en la topología.

- Planificación de paquetes, que responde a la dependencia de las condiciones del canal cuando planificamos paquetes en redes inalámbricas. Una gran variedad de disciplinas de planificación pueden ser usadas para encontrar el módulo de planificación de paquetes y el módulo de servicio. Actualmente, tenemos implementados una disciplina de servicio round-robin pesada basada en una implementación deficitaria de round-robin que ha sido extendida para compensar los casos de localizaciones dependientes de las condiciones del canal entre nodos móviles.

- MAC, que proporciona acceso al medio compartido guiado por calidad del servicio a servicios inalámbricos adaptativos o best effort. El esquema QoS INSIGNIA está diseñado para resultar transparente para cualquier protocolo de acceso al medio subyacente y está preparado para operar sobre tecnologías de capa multi-enlace en la capa IP. Sin embargo, el rendimiento de este esquema está fuertemente ligado a la provisión de QoS por controladores específicos de acceso al medio.

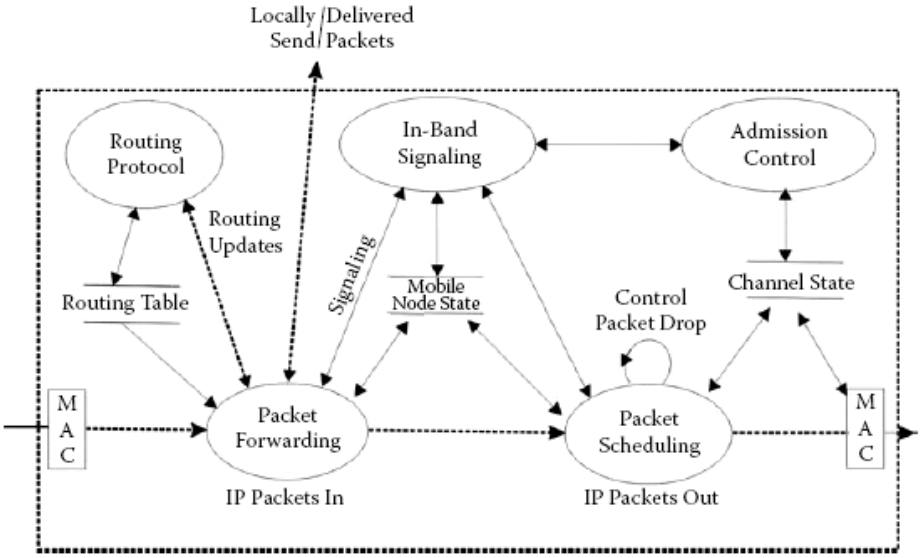


Figura 2.11.*Esquemas INSIGNIA.*

2.5.1.1. Sistema de señalización de INSIGNIA

El sistema de señalización INSIGNIA juega un papel importante en el establecimiento, adaptación, restauración y terminación de las reservas fin a fin. A continuación, describiremos el acercamiento de señalización en línea INSIGNIA. El sistema de señalización está diseñado para ser poco pesado en términos de cantidad de ancho de banda consumido por el control de la red y de ser capaz de reaccionar ante los elementos dinámicos de la red como la movilidad del servidor, degradación de los links gíreles y la conectividad intermitente de la sesión. Vamos a discutir primero los comandos del protocolo y luego los mecanismos del protocolo que lo implementan.

2.5.1.2. Comandos del protocolo INSIGNIA

Los comandos del protocolo son codificados usando el campo opcional IP y el indicador de servicios incluidos, tipo de carga, indicador de ancho de banda, y el campo requerido por el ancho de banda. Adoptando una opción IP INSIGNIA en cada cabecera IP, la complejidad de mantener la encapsulación del paquete dentro de la red es evitada. Estos comandos del protocolo mantienen la señalización de algoritmos incluyendo la reserva de flujo, restauración y adaptación de mecanismos. Los comandos del protocolo conducen los estados de operación del protocolo. Existen tres máquinas de estados finitos, para un servidor origen, router intermedio y servidor destino. Estas tres máquinas captan los eventos y acciones más significativas y las transiciones de estado resultantes. Usamos estas máquinas para ilustrar la dinámica del sistema de señalización INSIGNIA.

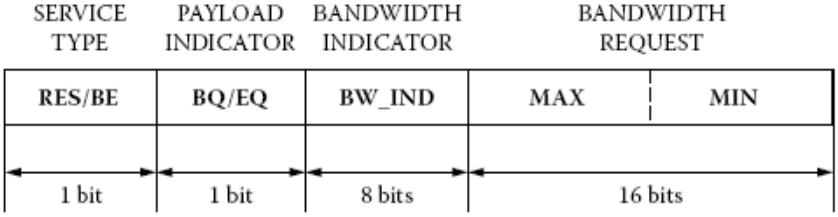


Figura 2.12. *Opción IP de INSIGNIA.*

2.5.1.2.1. Modo de servicio

Cuando un nodo origen quiere establecer una reserva rápida del nodo destino, establece en el modo de reserva a RES en la opción IP de INSIGNIA y envía el paquete al servidor destino. En la recepción de un paquete RES, intervienen nodos (funcionando como encaminadores) ejecuta el control de admisión para aceptar o denegar la consulta. Cuando un nodo acepta la consulta los recursos son comprometidos y los siguientes paquetes son programados dependiendo de este hecho. Por el contrario, si la reserva es denegada, los paquetes son tratados mediante el modo best-effort.

En el caso de que un paquete RES es recibido y no ha sido ubicado ningún recurso, el controlador de admisiones intentará hacer una nueva reserva. Esta condición sucede comúnmente cuando los flujos son reencaminados durante el tiempo de vida de una sesión en curso dependiente de la movilidad del servidor. Cuando el destino recibe un paquete RES, manda un informe QoS al nodo origen indicando que una reserva fin a fin ha sido establecida, y se produce una transición interna del estado de best-effort a reserva.

El modo reserva indica el nivel de servicios asegurados solicitados para soportar el servicio adaptable. La interpretación del modo servicio, que indica un paquete RES o BE, dependiente del tipo de carga y el indicador de ancho de banda. Un paquete con el modo servicio establecido en RES y el indicador de ancho de banda en MAX o MIN está intentando establecer servicios reservados máximos o mínimos, respectivamente. Los requisitos de ancho de banda del flujo son llevados en el campo consulta ancho de banda. Un paquete RES puede ser degradado a un servicio BE en caso de reencaminamiento o una insuficiencia de recursos a lo largo de una ruta nueva o ya existente. Nótese que un paquete BE no requiere reserva de recursos.

La opción IP también lleva una indicación del tipo de carga, que identifica si el paquete es una base QoS (BQ) o una mejora QoS (EQ). Usando el paquete estado, uno puede determinar que componente del flujo ha sido degradada. La recepción de un paquete BE/EQ/MIN o RES/BQ/MIN indica que el paquete mejora QoS ha sido degradado a un servicio best-effort. Monitorizando el paquete estado, el nodo destino puede emitir comandos de escala y caída al origen basados en el estado de la máquina destino.

Las máquinas origen, intermedia y destino soportan dos estados de reserva:

- Modo de máxima reserva, que provee reserva para flujos basados en QoS y paquetes mejora QoS. Este tipo de servicio requiere reservas fin a fin exitosas para unir el máximo número de necesidades de ancho de banda (ej. RES/EQ/MAX).
- Modo de mínima reserva, que provee reserva para flujos basados en QoS y envíuos best-efforts para las componentes de mejora QoS (si existen). Este modo servicio suele ocurrir cuando los flujos de máxima reserva sufren degradaciones en la red. Por ejemplo, flujos de máxima reserva pueden encontrarse servidores móviles y nodos que carezcan de recursos suficientes para soportar ambas componentes de base QoS y mejora QoS, produciendo una degradación de los paquetes mejora QoS a un sistema best-effort de envío (ej. BE/EQ/MIN)

2.5.1.2.2. Solicitud de ancho de banda

Las solicitudes de ancho de banda permiten a un origen especificar su Máximo (MAX) y Mínimo (MIN) ancho de banda requerido para servicios adaptables. Esto supone que el origen el modo de servicio RES. Un emisor puede simplemente especificar un mínimo y un máximo ancho de banda requerido. Para servicios adaptables, la base QoS (servicio min-reservado) es soportada por el mínimo ancho de banda, mientras que el máximo ancho de banda soporta el envío de la base y mejora QoS (servicio max-reservado) entre pares origen-destino. Los flujos son representados mediante la posesión de mínimos y máximos requisitos de ancho de banda. Esta caracterización es comúnmente usada para tráfico multirresolución (ej. MPEG audio y vídeo), los datos adaptables en tiempo real que tienen requisitos max-min discretos, y diferentes servicios que soportan la priorización de datos agregados en Internet.

2.5.1.2.3. Tipo de carga (payload)

El campo payloa indica el tipo de paquete que está siendo transportado. INSIGNIA soporta dos tipos de carga, BQ y EQ, lo cuales son reservados mediante un proceso distribuido de caontrol de admisión y reserva de paquetes. La semántica de los servicios adapatativos está relacionada con el tipo de carga y la disponibilidad de recursos (mejorar el QoS requiere que los requisitos máximos de ancho de banda se puedan cumplir a lo largo del camino entre la fuente y el destino). El significado de la base y la mejora de QoS es esppecífico de cada aplicación. Puede representar un simple esquema de prioridades entre paquetes, servicios diferenciales, etc. El proceso de adaptación puede obligar a que un flujo adapatativo se degrade cuando no hay suficientes recursos disponibles para soportar el ancho de banda máximo a lo largo de l camino o durante la restauración cuando el camino tiene insuficientes recursos. Por ejemplo, si sólo hay suficiente ancho de banda

2.5.1.2.4. Indicador de ancho de banda

Un indicador de ancho de banda desempeña un papel fundamental durante el proceso de reserva y de adaptación. Durante el establecimiento de la reserva, el indicador de ancho de banda indica la disponibilidad de recursos en los nodos intermedios a lo largo del camino entre los nodos fuente y destino. La recepción de un paquete de solicitud de creación con el bit indicador de ancho de banda con valor MAX indica que todos los nodos en la ruta tienen suficientes recursos para soportar el máximo ancho de banda solicitado. In cambio, un indicador de ancho de banda con valor MIN implica que al menos uno de los nodos intermedios entre la fuente y el destino desempeña el papel de cuello de botella y que, por lo tanto, no hay disponible suficiente ancho de banda para proporcionar el máximo requerido;

2.5.1.3. Funcionamiento del protocolo INSIGNIA

A continuación, vamos a revisar los principales mecanismos del protocolo y las máquinas de estados para las fuentes, nodos intermedios y nodos destino, como se muestra en la Figura . Los componentes de señalización incluyen establecimiento de reservas, informe de QoS, manejo de soft-state, restablecimiento de flujo y adaptación de flujo.

2.5.1.3.1. Establecimiento de reservas

Para establecer flujos adaptativos, los nodos fuente inician la reserva activando los flags correspondiente del campo IP options en el mensaje de datos antes de enviar paquetes reservation request hacia los nodos destino. Un paquete reservation request se caracteriza por tener el modo de servicio marcado como RES, el payload marcado como BQ/EQ, y el indicador de ancho de banda como MAX/MIN y requisitos de ancho de banda válidos. Los paquetes de reserva atraviesan los nodos intermedios ejecutando los módulos de control de admisión, asignación de recursos, y establecimiento del estado de flujo en todos los nodos intermedios entre la fuente y el destino, como se ilustra en la Figura 2.13. Un nodo fuente continúa enviando paquetes de reserva hasta que el nodo destino completa la fase de configuración de reservas informando al nodo fuente del

estado de la fase de establecimiento del flujo usando informes de QoS como en la Figura 2.13.

El establecimiento de un flujo adaptativo se muestra en la Figura . Un nodo fuente (M_s) solicita la asignación máxima de recursos, y el nodo M_1 ejecuta el control de admisión tras la recepción del paquete de reserva. Los recursos son asignados si se encuentran disponibles, y el paquete de reserva es enviado al siguiente nodo (M_2). Este proceso es repetido hasta que el paquete de reserva alcanza el nodo destino M_D. El nodo destino determina el estado de la asignación de recursos comprobando el paquete state (modo de servicio, tipo de payload, indicador de ancho de banda). El mecanismo de informa de QoS se usa para informar a la fuente del estado de la reserva en la ruta. En lo que concierne al nodo destino, la fase de reserva se termina en cuanto recibe el primer paquete RES. En el ejemplo vemos que sólo se soporta el anchote banda mínimo entre M2 y M3, y que los nodos siguientes permiten asignar recursos para el máximo.

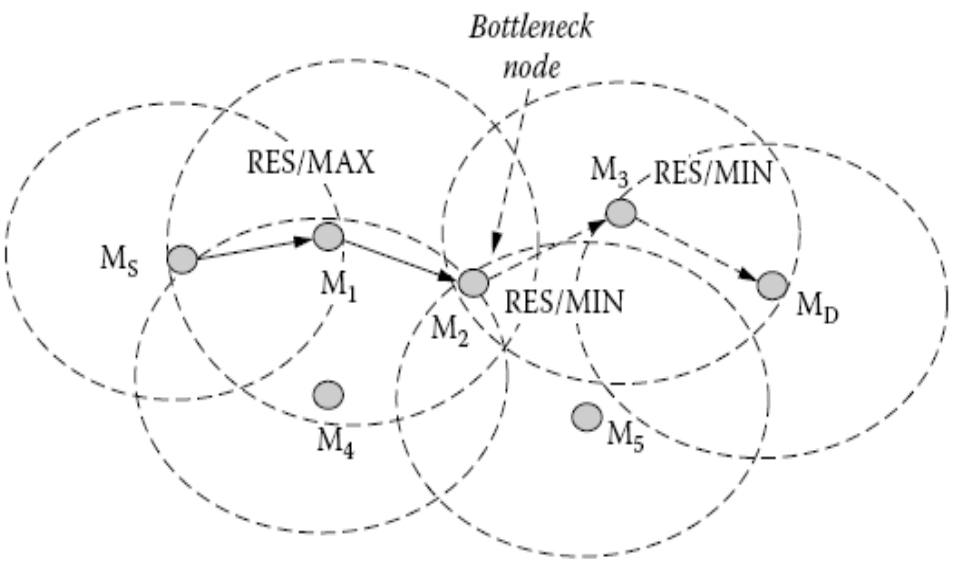


Figura 2.13. *Reserva de flujo con servicio adaptativo.*

Cuando una reserva es recibida por el nodo destino, el módulo de señalización comprueba el estado del establecimiento del flujo. El estado de la fase de reserva se determina inspeccionando el flag del campo IP options de modo de servicio, que debe estar marcado como RES. Si el indicador de anchote banda está marcado xomo MAX, esto implica que todos los nodos entre la fuente y el destino han asignado satisfactoriamente los recursos necesarios para poder operar en el modo max-reserved. Por otro lado, si el indicador de ancho de banda muestra MIN, indica que sólo se puede proporcionar el QoS básico, es decir, el modo min-reserved. En este caso, todos los paquetes de reserva con un payload EQ recibidos en el destino verán su nivel de servicio modificado de RES a BE por el nodo que actúa de cuello de botella. Como resultado, existirán reservas parciales entre la fuente y el nodo cuello de botella.

En el caso de reservas parciales, los recursos permanecen reservados entre la fuente y el cuellote botella hasta que son liberados explícitmanete. La liberación de recursos reservados parcialmente puede ser iniciada por la fuente como reacción durante

la fase de reserva o como parte del proceso de adaptación donde el destino puede mandar un comando de borrado al nodo fuente. Esto tendrá el efecto de borrar cualquier reserva parcial Una aplicación puede elegir no borrar las reservas parciales, confiando en que el ancho de banda llegará a estar disponible en el cuello de botella.

Observemos que si se ha establecido una reserva para el estado de reserva máxima y se reciben de manera constante paquetes RES/BQ/MIN, la máquina determina que los paquetes de mejora de QoS han sido degradados y cambia al estado de reserva mínimma. Degradaciones de este tipo pueden ocurrir en los nodos intermedios debido a la falta de recursos para soportar la nueva reserva, o porque un flujo que está en camino se degrada por el redireccionamiento o la insuficiencia de recursos en el nuevo o existente camino. La información de estado mantenida en el destino permite descubrir cual de estas condiciones ha ocurrido.

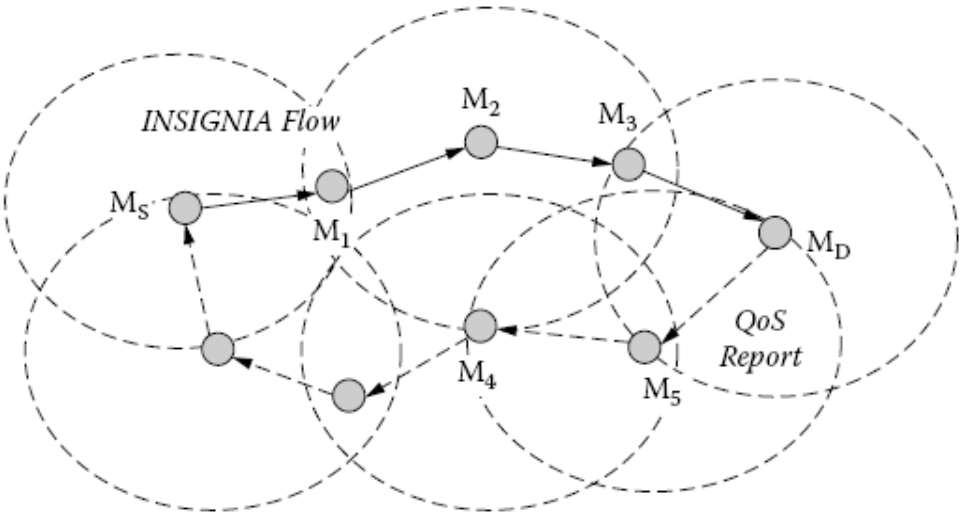


Figura2.14 . *Notificación de QoS.*

2.5.1.3.2. Notificación de QoS

La notificación de QoS se usa para informar a los nodos fuentes del estado de transmisión del flujo. Los nodos destino monitorizan activamente los flujos en servicio inspeccionando la información del estado (indicadores de ancho de banda) y midiendo el QoS efectiv (pérdida de paquetes, retardo, productividad). Informes de QoS son enviados los nodos fuente para completar la fase de reserva y, de forma periódica, para manejar las adaptaciones "extremo a extremo". Los informes de QoS no tienen que viajar de por el camino inverso hacia la fuente. Normalmente toman una ruta alternativa a través de la red ad hco como se muestra en la Figura .Así que los informes QoS son generados básicamente de forma periódica de acuerdo con la sensibilidad de las aplicaciones a la calidad de servicio. Los informes QoS son enviados inmediatamente cuando así se requiere (normalmente acciones relacionadas con la adaptación).

En el caso en el que sólo se puedan soportar paquetes BQ, como en el caso con modo min-reserved, el sistema de señalización de la fuente cambia el modo de servicio de los paquetes BQ de RES a BE, con todos los paquetes degradados enviados con best effort. Algunas reservas parciales que pueden existir entre los nodos fuente y destino

2.5.1.3.3. Restablecimiento del flujo

Los flujos son frecuentemente redireccionados dentro del tiempo de vida de las sesiones de transmisión debido a la movilidad de los teminales. El objetivo del restablecimiento de flujo es recuperar las reservas tan rápido y eficiente como se

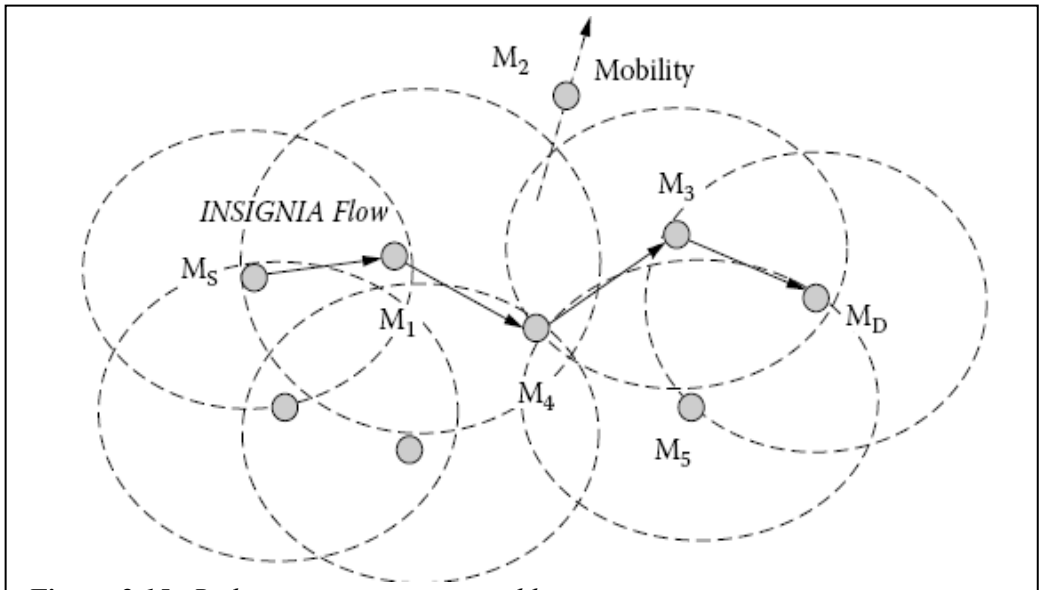


Figura 2.15 . *Redireccionamiento y restablecimiento.*

aposible. Redireccionar flujos activos involucra al protocolo de enrutamiento (para determinar una nueva ruta), el control de admisión y la reserva de recursos para nodos que van a pertenecer al nuevo camino. Los procedimientos de restauración llaman también a los procedimientos de eliminación para borrar el estado del flujo en los nodos que formaban el camino antiguo. En el caso ideal, el restablecimiento de flujos puede acometerse dentro de la duración de unos pocos paquetes consecutivos dado que existe una ruta alternativa almacenada. A este tipo de restablecimiento le llamamos restablecimiento inmediato. Si no existe una ruta alternativa, el rendimiento del algoritmo de restauración está ligado a la velocidad a la que el algoritmo de enrutamiento puede descubrir un nuevo camino.

Como se muestra en la Figura , el dinamismo de la red provoca el redireccionamiento y la degradación del servicio. En este ejemplo, el nodo móvil M_2 se aleja del radio de contacto y se pierde la conectividad. El nodo emisor, M_1, interactúa con el protocolo de enrutamiento y envía el paquete por la nueva ruta. El sistema de señalizaciónen el router intermedio M_4 recibe paquetes e inspecciona su tabla soft-state. Si no existe una reserva para los nuevos paquetes que llegan, el módulo de señalización invoca al control de admisión e intenta asignar recursos para el flujo. Observamos que cuando los paquetes redireccionados llegan al nodo M_3, detecta que existe una reversa para dichos paquetes, por lo que se reincorporan a su antiguo camino.Los temporizadores soft-state aseguran que el estado de flujo está todavía intacto en M_3 y los estados a través del camino antiguo (M_2) son liberados de una manera eficiente.

Cuando un flujo adaptativo es redireccionado a un nodo donde los recursos no están disponibles, el flujo es degradado a un servicio best-effort. En consecuencia, los nodos descendientes reciben estos paquetes degradados y no intentan asignar recursos o actualizar el estado de reserva asociado con el flujo. En este caso, este estado asociado con un flujo caduca y los recursos son liberados. Una reserva debe ser restaurada si los

recursos llegan a estar disponibles en el nodo cuello de botella (M_4 en Figura) o si un redireccionamiento más lejano permite completar el restablecimiento. A este tipo de restablecimiento lo llamamos restablecimiento degradado. Un flujo puede permanecer degradado durante la duración de la sesión., sin ser restablecido, lo cual se denomina degradación permanente. La componente de QoS de un flujo adaptativo debe ser degradada a un servicio best-effort durante el restablecimiento de ruta si la ruta alternativa sólo puede soportar el ancho de banda mínimo requerido. Si la degradación del paquete de mejora QoS persiste, puede causar la interrupción de los servicios y provocar que el nodo móvil de destino invoque su procedimiento de adaptación para reducir o eliminar los paquetes con bastante tiempo de vida con baja calidad. Los mecanismos de actualización localizados en los nodos destino están capacitados para responder a cambios en la disponibilidad de recursos de la red a través de las acciones "amentar", "reducir" y "eiliminar" en respuesta a las condiciones de la red.

Durante el proceso de restablecimiento, el esquema INSIGNIA no favorece redireccionar flujos sobre flujos existentes y favoreciendo algunos criterios de equidad. En este sentido, INSIGNIA permite la introducción de fluctuaciones de servicios adicionales de flujos existentes para soportar el restablecimiento de rutas redireccionadas. Como resultado de esta política, el control de admisión sólo rechaza o elimina algún flujo redireccionado cuando los recursos disponibles son insuficientes a lo largo del nuevo camino.

El esquema de QoS INSIGNIA soporta tres tipos de restablecimiento:

- Restablecimiento inmediato, el cual ocurre cuando un flujo redireccionado recupera inmediatamente su reserva original; esto es, un flujo con modo max-reserved es restaurado a un flujo en modo max-reserved, y un flujo con modo min-reserved será restaurado a un flujo en modo min-reserved.

- Restablecimiento degradado, el cual ocurre cuando un flujo es redireccionado es degradado por un período (T) antes de recuperar su reserva original. Pueden darse de dos formas diferentes:
    - Un flujo en modo max-reserved opera en modo min-reserved o best-effort, y recobra su servicio en modo max-reserved después de un intervalo.
    - Un flujo en modo min-reserved opera en modo best-effort y recobra su modo de servicio original min-reserved después de un intervalo.

- Degradación permanente, el cual ocurre cuando el flujo redireccionado recupera su reserva original.

La Figura muestra los cambios topológicos que ocurre después de la redirección basado en la topología original de la Figura . Después de redirigir el enlace M4, M5 puede soportar sólo servicios best-effort. Este tipo de restablecimiento representa o bien un restablecimiento degradado o una degradación permanente. En este escenario, el nodo destino limpia la reserva parcial entre los nodos móviles M5 y M4 enviando un comando de adapación "drop" a la fuente. El proceso de restablecimiento puede ser inmediato o retrasado. La adaptación es específica de la aplicación, donde la aplicación puede elegir responder a las condiciones de la red y el QoS proporcionado.

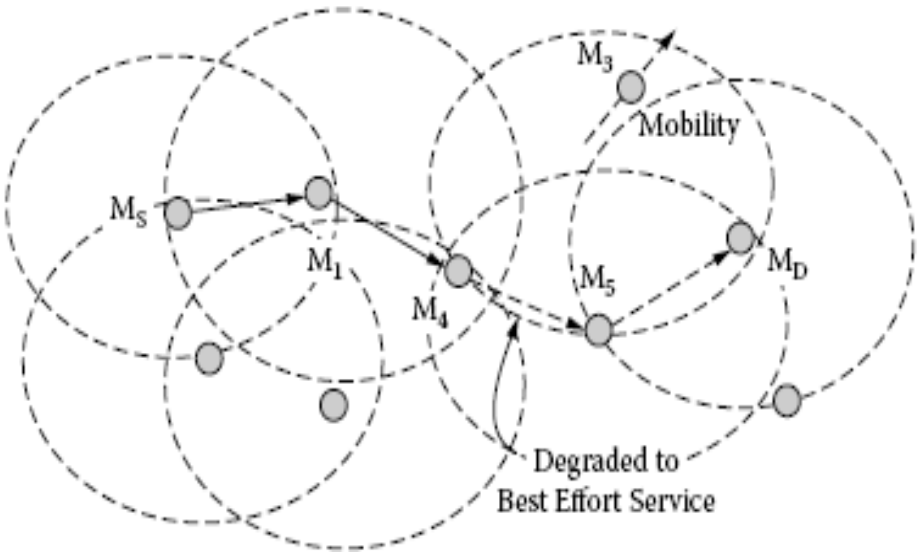


Figura 2.16. *Redireccionamiento y degradación.*

2.5.1.3.4. Adaptación del flujo

El esquema de QoS INSIGNIA monitoriza activamente el dinamismo de la red y adapta los flujos basándose en la respuesta a los cambios observados o a una política de adaptación proporcionada por el usuario. La calidad de recepción del flujo es monitorizada por el nodo destino y basado en una política de adaptación específica de la aplicación. Se realizan las acciones para adaptar el flujoa unas condiciones observadas. La toma de decisiones está condicionada a lo que está programado en la política de adaptación del usuario. Por ejemplo, una política de adaptación podría ser mantener el nivel de servicio ante la degradación d elas condiciones o reducir los flujos adaptativos a su QoS básico en respuesta a la degradación de las condiciones. Otros aspectos de la política podría ser escalar siempre los flujos adaptativos cuando los recursos estén disponibles. La aplicación es libre de programar su propia política de adaptación, la cual es ejecutada por INSIGNIA a través de la interacción entre los nodos destino y fuente.

INSIGNIA proporciona un conjunto de niveles de adaptación que pueden ser seleccionados. Normalmente, un flujo adaptativo opera con ambos su base y sus componentes mejorados siendo transportados con reserva de recursos. Reducir el flujo depende de la política de adaptación seleccionada. El flujo puede ser reducido a su QoS base enviando paquetes de mejora de QoS en modo best-effort.

2.5.2. Cross-Layer Design for Data Accessibility

La arquitectura del Cross-Layer Desgin se muestra en la Figura . Las capas de aplicación, enlace de datos y de red comparten información para conseguir una mayor calidad en el acceso a la información. El sistema hace uso de la replicación de datos para evitar el problema de la pérdida de información cuando ocurre una partición en la red. El visionado de mapas y la mensajería son dos ejemplos de aplicaciones que se muestran en la figura.

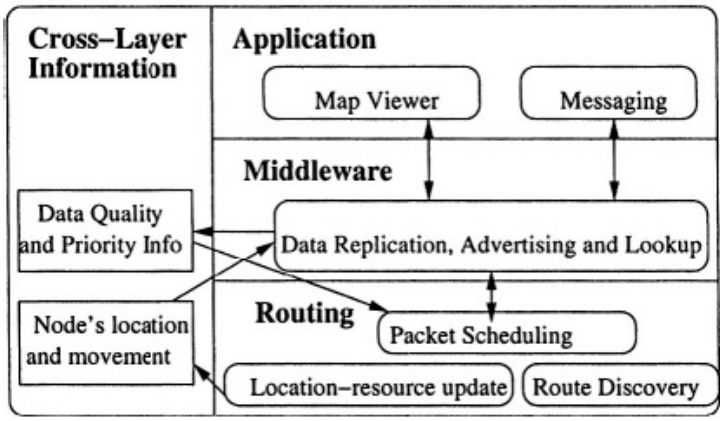


Figura 2.17. *Estructura del Croos Layer Design for Data Accesibility*

La capa de red emplea un protocolo de enrutamiento basado en la predicción de la localización (predictive location-based routing protocol). Usa las coordenadas geométricas de cada nodo y su patrón de movimiento para el descubrimiento de rutas y su mantenimiento. El módulo de actualización de localización de recursos envía periódicamente por broadcast mensajes que contienen la localización de los nodos e información sobre sus recursos a otros nodos de la red. La capa de red reacciona al deterioro del rendimiento de la ruta recalculándola.

La capa de enlace de datos implementa un servicio de accesibilidad a datos que ayuda a la aplicación anunciar y compartir datos con otros usuarios de la red. Los datos son accesibles en dos pasos. En el primer paso se obtiene la información sobre la disponibilidad de los datos y presentada a la capa de aplicación. El parámetro de QoS que interesa es la tasa de éxito en el acceso a datos. En el segundo paso, la capa de enlace de datos recupera los datos de un nodo remoto con unos ciertos requisitos a nivel de aplicación, como la calidad de los datos y su límite. La capa de enlace de datos traduce los requisitos de la capa de aplicación a parámetros de QoS de la red, como ancho de banda y retardo. A continuación establece una ruta con dichos parámetros. Para preservar la red de violaciones de QoS, la capa de enlace de datos es avisada por el protocolo de enrutamiento si no es capaz de establecer o mantener la ruta. La capa de enlace de datos se debe adaptar a los recursos disponibles.

2.5.3. Intelligent Optimization Self-Regulated Adjustment (INORA)

INORA es un mecanismo de soporte de QoS que hace uso de la señalización in-band de INSIGNIA y del protocolo de enrutamiento para MANETs TORA (descrito en el capítulo 6). INORA representa un esquema de señalización de QoS que resulta de fusionar arias técnicas. La idea está basada en la propiedad de TORA de proporcionar varias rutas entre un origen y un destino dados. Aunque INSIGNIA no necesita ninguna ayuda de la red para redirigir el flujo a través de las rutas que son capaces de proporcionar las garantías de QoS requeridas, INORA proporciona feedback (per hop) al protocolo de enrutamiento para dirigir el flujo por la ruta que satisface los requisitos de QoS del flujo. Sin ninguna duda, el enfoque de señalización y enrutamiento QoS "ligeramente unidos" es un esquema muy prometedor, y las deficiencias de INORA son en su mayoría, las deficiencias de INSIGNIA. Sin embargo, el interfaz para el acceso al

enrutamiento de la señalización debe ser lo más genérica posible para garantizar la portabilidad.

TORA opera creando un Grafo Acíclico Dirigido (DAG) con raíz en el destino. El DAG es extremadamente útil en este esquema porque proporciona múltiples rutas desde el emisor hasta el destino. Se usa esta estructura de enrutamiento para dirigir el flujo por rutas que puedan proporcionar recursos al flujo de acuerdo con sus requisitos de QoS. INORA puede dividirse en dos esquemas:

- Esquema coarse-feedback
- Esquema fine-feedback

2.5.3.1. Esquema Coarse-Feedback

El funcionamiento del esquema coarse-feedback de INORA se puede describer a través del siguiente ejemplo. Consideramos un flujo QoS iniciado por un nodo 1 y con destino un nodo 5.

- Sea el DAG creado por TORA el mostrado en la Figura 2.18. Sea 1 → 2 → 3 → 4 → 5 sea el camino escogido por TORA.

- INSIGNIA intenta reservar recursos para el flujo QoS a través del camino. El nodo 4 es el primer nodo donde el control de admisión para el flujo falla. El nodo 4 envía un mensaje de Admisión Control Failure (ACF) de fuera de banda al nodo anterior.

- El nodo 3 se da cuenta de que el siguiente salto, el nodo 4, no puede soportar los requisitos de este flujo, así que prueba con otro nodo descendente, nodo 6, proporcionado por TORA (Figura 2.18).

- Si el nodo 6 es capaz de admitir el flujo, el flujo tiene reservados los recursos requeridos a lo largo del camino. El nuevo camino sería 1 → 2 → 3 → 6 → 5 (Figura 2.18).

- Si el nodo 6 es incapaz de admitir el flujo, envía un mensaje ACF al nodo 3 (Figura 2.18).

- El nodo 3 se da cuenta de que ninguno de los nodos descendentes proporcionados por TORA puede satisfacer los requisitos de QoS del flujo. Por tanto, envía a su predecesor (nodo 2) un mensaje ACF indicando que ninguno de sus descendientes puede dar soporte al flujo (Figura 2.18).

- El nodo 2 intenta ahora con sus otros descendientes vecinos crear un camino que satisfaga los requisitos del flujo (Figura 2.18).

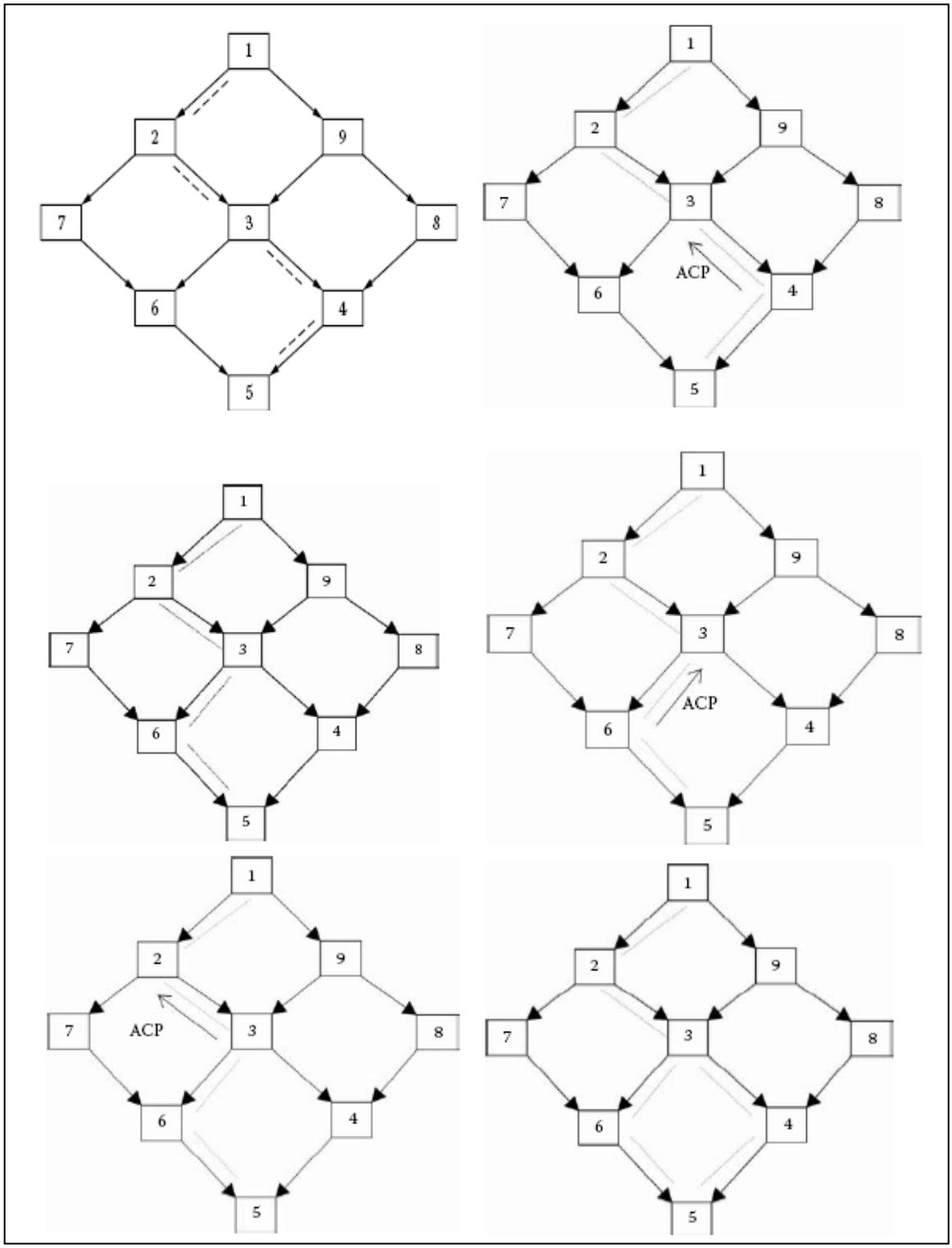


Figura 2.18. *Funcionamiento de INORA con Coarse Feedback.*

Hay que tener en cuenta de que como resultado de la aplicación de este esquema, es posible que distintos flujos entre el mismo par fuente y destino tomen diferentes rutas, por ejemplo, para ir del nodo 1 al nodo 5, el flujo 1 puede tomar el camino 1 → 2 → 3 → 4 → 5 y el flujo 2 tomar el camino 1 → 2 → 3 → 6 → 5.

2.5.3.2. Esquema Class-Based Fine Feedback

En este esquema, dividimos el intervalo (Bwmin, BWmax) en N clases, donde BWmin es el ancho de banda mínimo requerido por un flujo con QoS y BWmax es el ancho de banda máximo requerido por un flujo con QoS. El campo IP options en la cabecera IP, que lleva la información de INSIGNIA, lleva ahora un campo class adicional. Este campo significa la cantidad de ancho de banda que ha sido asignado al flujo a lo largo del camino. El modo de operación de este protocolo se muestra en les siguiente ejemplo.

Consideramos que un flujo con QoS se ha iniciado en el nodo 1 y tiene como destino el nodo 5, con un requerimiento mínimo de ancho de banda BWmin y un requerimiento máximo de ancho de banda de BWmax. Sea el flujo admitido con clase (m < N) m en el nodo 1.

- Sea el DAG creado por TORA como el mostrado en la Figura 2.19. Sea 1 2 3 4 5 el camino escogido por el protocolo de enrutamiento.
- INSIGNIA intenta establecer reservas sofá-state para el flujo QoS a lo largo del camino.
- El nodo 2 es capaz de admitir el flujo con clase m como fue solicitado por su nodo upstream, el nodo 1.
- Suponemos que el nodo 3 ha admitido el flujo con clase l, pero no ha sido capaz de asignar el ancho de banda de clase m (l < m), como solicitó su antecesor el nodo 2.
- Nodo 3 envía un mensaje Admisión Report (AR) conteniendo la clase asignada AR(l) al nodo previo upstream, el nodo 2. este mensaje indica la capacidad de asignar la clase l de ancho de banda al flujo solicitado.
- Nodo 2 divide el flujo con una relación de l a (m - l) y envía el flujo al nodo 3 y al nodo 7, respectivamente, en esa proporción. Esto significa que el flujo de clase m se ha dividido en dos flujos de clase l y clase (m - l) que son enviados al nodo 3 y al nodo 7 respectivamente.
- Suponemos que el 7 es incapaz de asignar la clase (m - l) como solicitó su antecesor, el nod 2, y sólo es capaz de asignar la clase n < (m - l). el nodo 7 envía un mensaje AR(n) al nodo 2. Figura 2.19.
- El nodo 2, detecta que sus nodos descendientes no son capaces de proporcionar la clase m, que ha sido solicitada, así que informa a su antecesor de su capacidad para asignar una clase (l + n < m) l + n enviando un mensaje AR. Figura 2.19.
- El nodo 1 intenta ahora encontrar otro nodo descendiente que pueda ser capaz de asignar una clase m – (l + n). Figura 2.19.

Como vemos, cuando un nodo es incapaz de admitir un flujo, o bien porque no puede asignar el ancho de banda mínimo Bimini requerido por el flujo o debido a la conestión en un nodo, envía mensajes ACF como en el esquema coarse-feedback. Por tanto, el esquema fine-feedbackk incluye las características del esquema coarse-feedback. El esquema fine-feedback, como el esquema coarse-feedback, intenta primero encontrar una ruta QoS que pueda dar el ancho de banda solicitado localmente. La búsqueda llega a ser más global si no es capaz de encontrar localmente la ruta QoS que da la clase requerida. Puesto que es peor nada que el que un flujo pueda ser dividido, se envían los paquetes por diferentes rutas hasta el destino. Esto puede hacer que se reciban los paquetes desordenados en el destino. Las aplicaciones de tiempo real con requisitos de QoS usan Real-Time Transport Protocol (RTP) como el protocolo de

transporte. RTP reordena los paquetes. Si se usa TCP como protocolo de transporte, la llegada de paquetes fuera de orden pueden disparar los mecanismos de control de congestión de TCP. El efecto del envío de paquetes desordenados con TCP debe ser aún

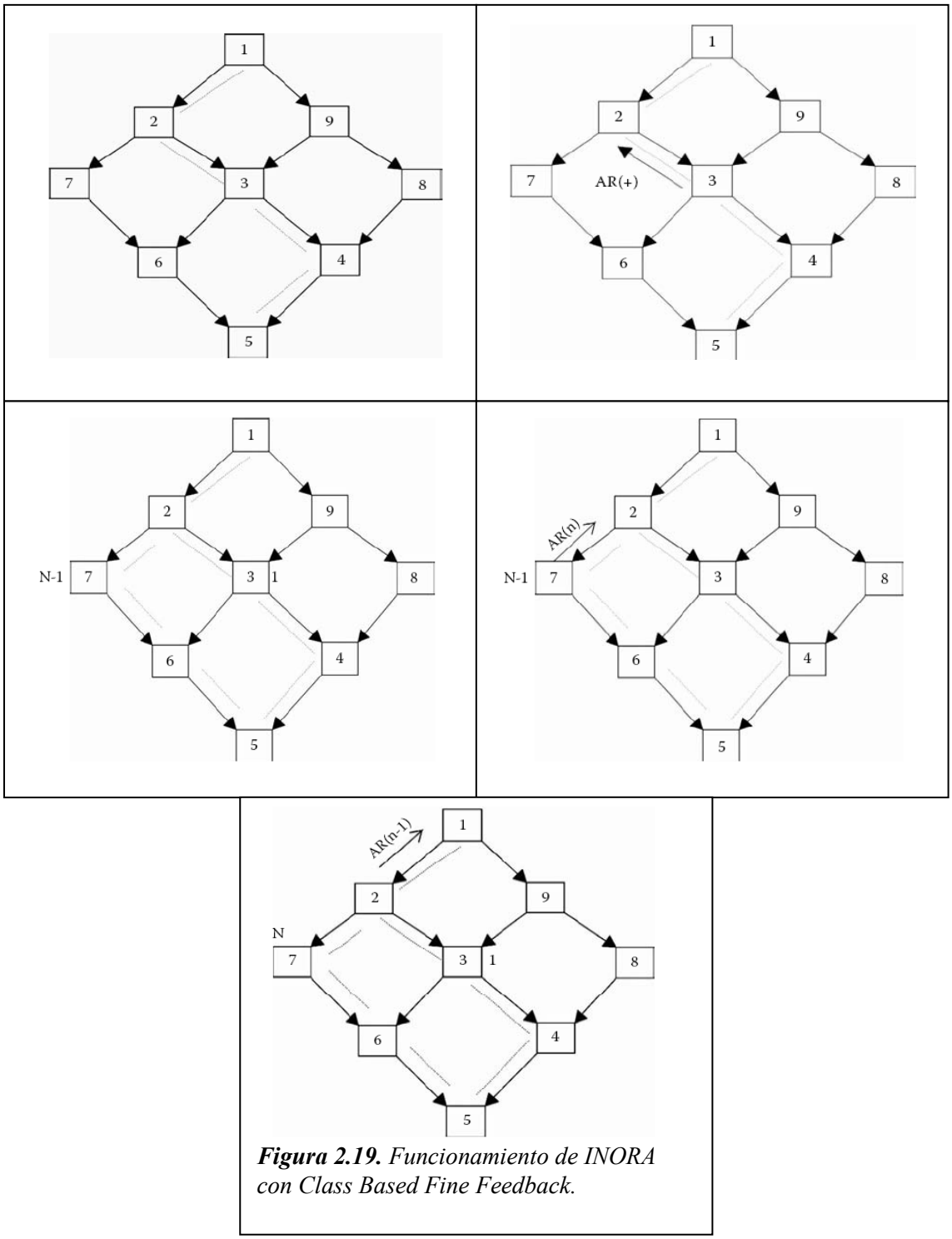


Figura 2.19. *Funcionamiento de INORA con Class Based Fine Feedback.*

estudiado.

Evaluamos el rendimiento de los esquemas INORA observando el retardo "extremo a extremo" de los paquetes y la sobrecarga de mensajes de control. Vemos que esquema INORA con coarse-feedback presenta unos valores de retardo menores que INSIGNIA y TORA operando sin feedback. El esquema INORA con fine-feedback

mejora al esquema INORA con coarse-feedback. Esto es porque los esquemas INORA con feedback intentan encontrar caminos en los cuales se puedan conceder las reservas de ancho de banda solicitadas para los flujos QoS. El esquema fine-feedback trabaja de una manera de grano más fino comparado con el esquema coarse-feedback. Esto hace, junto con los datos empíricos, que se pueda asegurar que el esquema fine-feedback aporta mejores rendimientos que el esquema-feedback.

2.6. Conclusiones

En este capítulo hemos estudiado los aspectos relacionados con QoS n varias capas de redes ad-hoc. La capa física y la capa MAC son las principales responsables de QoS en un solo enlace. El DCF y la funcionalidad PCF de 802.11 está siendo extendida en la extensión QoS llamada 802.11e. El PCF y los protocolos 802.11 están especialmente diseñados para dar soporte a redes QoS de un solo salto. La capa de enrutamiento es responsable de calcular y mantener rutas fin a fin de multi salto en QoS. CEDAR [16] y Enrutamiento Basado en Ticket [20] son dos protocolos QoS de enrutamiento propuesto para redes as-hoc. Desde que QoS necesita ascender por la capa de aplicación, los requisitos de QoS para los valores aceptables de sus métricas son especificados por la aplicación. Las consultas QoS puede que tengan que viajar desde las capas de red hasta las físicas. Las aplicaciones pueden querer ser notificadas en caso de que los requisitos QoS no puedan ser satisfecho debido a los cambios en las condiciones de la red. La aplicación debe ser capaz de (re)negociar uno requisitos QoS diferentes y adaptarse a ellos.
QoS es actualmente un área de investigación activa en redes ad-hoc. Este capítulo ha cubierto algunos de los temas principales en la investigación de redes ad-hoc.

Aun así hay numerosos caminos que requieren una mayor exploración para diseñar un QoS compatible con redes ad-hoc. Vamos a destacar brevemente algunos de esos aspectos:

- Arquitectura energética eficiente en QoS: las redes ad-hoc están condicionadas por la energía, dado que están compuestas por dispositivos portátiles con una batería limitada.

- Los niveles de tolerancia medidos en QoS: el protocolo CEDAR y el basado en ticket intentan calcular rutas QoS. Estos protocolos no aportan grandes garantías en ninguna métrica QoS. El origen puede especificar el nivel de tolerancia QoS y entonces la red dará soporte a la consulta dependiendo de los niveles de tolerancia.

- Capa MAC sincronizada multisalto: para paquetes que realizan múltiples saltos, el QoS fin a fin es una función de mediciones QoS en cada link intermedio. Las propiedades QoS fin a fin pueden ser mejoradas mediante el diseño de una capa MAC que se coordine con otros nodos intermedios multisalto.

- Extendiendo PCF y 802.11e para redes ad-hoc: Ambas soluciones requieres el punto coordinador (o el punto de acceso) para decidir el esquema de la transmisión. Dado que no hay un control centralizado en una red ad-hoc, aunque

esta funcionalidad requiera ser creada de forma distribuida o deban hacerse otros cambios en estos protocolos para usarlos en redes ad-hoc.

Descubrimos que QoS es una componente inherente de una red ad-hoc y que hay varias cuestiones sin resolver que necesitan ser diseccionados a redes ad-hoc QoS diseñadas y activas.

3. AUTOCONFIGURACIÓN

Una MANET es una red compuesta de routers MANET, cada unos de los cuales posee al menos una interfaz MANET. En este capítulo vamos a definir los objetivos de los mecanismos de autoconfiguración en MANET, con respecto a los parámetros necesarios para la identificación IP. Concretamente, nos centraremos en los requisitos para:

- Autoconfigurar los interfaces MANET con direcciones IPv6.
- Asignación automática de prefijos IPv6 a los routers MANET.

3.1. Categorías de MANETs

La autoconfiguración de direcciones IP en interfaces MANET y la asignación de prefijos para routers IP pueden ser empleadas en diferentes escenarios. Vamos a describir los diferentes escenarios en los que se pueden asignar direcciones y prefijos IP mediante las soluciones de autoconfiguración existentes para MANETs.

3.1.1. MANETs subordinadas

Una MANET subordinada es una MANET que está conectada al menos a una red externa N que impone una jerarquía específica de direcciones a la MANET. En una MANET subordinada, esta jerarquía de direcciones obliga al uso de prefijos específicos para las comunicaciones entre nodos de la AMNET y los nodos pertenecientes a la red N. Por ejemplo, estos prefijos necesitan ser topológicamente correctos. Esto se pude conseguir siendo asignados dentro de un prefijo p::, sobre el cual el punto de unión con la red N tiene autoridad.

3.1.1.1. Ejemplos de MANETs subordinadas

Un típico ejemplo de MANET subordinada es una MANET que es parte de Internet, el cual obliga al uso de direcciones IP topológicamente correctas para poder comunicarse correctamente a través de Internet. Por ejemplo redes mesh públicas, con accesos WLAN fijos distribuidos a lo largo de un espacio participando en una MANET de usuarios móviles y actuando como routers periféricos.

Otro ejemplo es el área de cobertura de una red inalámbrica WAN, donde uno o más routers MANET son conectados a Internet a través de tecnologías como UMTS o WiMAX.

Redes de comunicaciones de vehículos conectados a una infraestructura externa debe ser considerada otro tipo de MANET subordinada.

3.1.2. MANETs autónomas

Las MANETs autónomas son MANETs sobre las que no existen redes externas que impongan una jerarquía de direcciones.

3.1.2.1. Ejemplos de MANETs autónomas

Un ejemplo de MANET autónoma son las redes establecidas en áreas donde no existen infraestructuras o no son apropiadas. Por ejemplo, comunicaciones entre coches para compartir información sobre el tráfico y la seguridad, comunicación de emergencia entre los miembros de un equipo de rescate en una situación catastrófica, compartición de archivos en una sala de conferencias o aula.

3.2. Objetivos de la autoconfiguración de MANETs

El objetivo de la autoconfiguración de redes MANETs es proporcionar mecanismos que permitan a cada router MANET:

- Configurar direcciones Ipv6 que sean únicas dentro de la MANET, en su interfaz MANET.
- Asignar prefijos IPv6 que sean diferentes de los prefijos asignados a otros routers de la MANET.
- Mantener, dentro de la MANET, la unicidad de las direcciones y de los prefijos asignados (incluyendo el caso de la mezcla de redes)
- Asignar Prefijos topológicamente correctos en los escenarios de MANETs subordinadas.

3.3. Aplicabilidad de las soluciones de autoconfiguración standard

Una vez hemos definido que requisitos deben cumplir nuestros mecanismos de autoconfiguración, vamos a revisar las soluciones existentes para otros tipos de redes y a discutir su aplicabilidad a MANETs.

3.3.1. DHCP

DHCP significa Protocolo de configuración de host dinámico . Es un protocolo que permite que un equipo conectado a una red pueda obtener su configuración (principalmente, su configuración de red) en forma *dinámica* (es decir, sin intervención particular). Sólo tiene que especificarle al equipo, mediante DHCP, que encuentre una dirección IP de manera independiente. El objetivo principal es simplificar la administración de la red.

El protocolo DHCP sirve principalmente para distribuir direcciones IP en una red, pero desde sus inicios se diseñó como un complemento del protocolo BOOTP

(Protocolo Bootstrap), que se utiliza, por ejemplo, cuando se instala un equipo a través de una red (BOOTP se usa junto con un servidor TFTP donde el cliente encontrará los archivos que se cargarán y copiarán en el disco duro). Un servidor DHCP puede devolver parámetros BOOTP o la configuración específica a un determinado host.

3.3.1.1. Funcionamiento

Primero, se necesita un servidor DHCP que distribuya las direcciones IP. Este equipo será la base para todas las solicitudes DHCP por lo cual debe tener una dirección IP fija. Por lo tanto, en una red puede tener sólo un equipo con una dirección IP fija: el servidor DHCP.

El sistema básico de comunicación es BOOTP (con la trama UDP). Cuando un equipo se inicia no tiene información sobre su configuración de red y no hay nada especial que el usuario deba hacer para obtener una dirección IP. Para esto, la técnica que se usa es la transmisión: para encontrar y comunicarse con un servidor DHCP, el equipo simplemente enviará un paquete especial de transmisión (transmisión en 255.255.255.255 con información adicional como el tipo de solicitud, los puertos de conexión, etc.) a través de la red local. Cuando el DHCP recibe el paquete de transmisión, contestará con otro paquete de transmisión (no olvide que el cliente no tiene una dirección IP y, por lo tanto, no es posible conectar directamente con él) que contiene toda la información solicitada por el cliente.

Se podría suponer que un único paquete es suficiente para que el protocolo funcione. En realidad, hay varios tipos de paquetes DHCP que pueden emitirse tanto desde el cliente hacia el servidor o servidores, como desde los servidores hacia un cliente:

- **DHCPDISCOVER** (para ubicar servidores DHCP disponibles)
- **DHCPOFFER** (respuesta del servidor a un paquete DHCPDISCOVER, que contiene los parámetros iniciales)
- **DHCPREQUEST** (solicitudes varias del cliente, por ejemplo, para extender su concesión)
- **DHCPACK** (respuesta del servidor que contiene los parámetros y la dirección IP del cliente)
- **DHCPNAK** (respuesta del servidor para indicarle al cliente que su concesión ha vencido o si el cliente anuncia una configuración de red errónea)
- **DHCPDECLINE** (el cliente le anuncia al servidor que la dirección ya está en uso)
- **DHCPRELEASE** (el cliente libera su dirección IP)
- **DHCPINFORM** (el cliente solicita parámetros locales, ya tiene su dirección IP)

El primer paquete emitido por el cliente es un paquete del tipo DHCPDISCOVER. El servidor responde con un paquete DHCPOFFER, fundamentalmente para enviarle una dirección IP al cliente. El cliente establece su configuración y luego realiza un DHCPREQUEST para validar su dirección IP (una solicitud de transmisión ya que DHCPOFFER no contiene la dirección IP) El servidor simplemente responde con un DHCPACK con la dirección IP para confirmar la asignación. Normalmente, esto es suficiente para que el cliente obtenga una

configuración de red efectiva, pero puede tardar más o menos en función de que el cliente acepte o no la dirección IP.

Para optimizar los recursos de red, las direcciones IP se asignan con una fecha de inicio y de vencimiento para su validez. Esto es lo que se conoce como "**concesión**". Un cliente que detecta que su concesión está a punto de vencer, puede solicitarle al servidor una extensión de la misma por medio de un DHCPREQUEST. Del mismo modo, cuando el servidor detecta que una concesión va a vencer, enviará un DCHPNAK para consultarle al cliente si desea extenderla. Si el servidor no recibe una respuesta válida, convertirá la dirección IP en una dirección disponible.

Esta es la efectividad de DHCP: se puede optimizar la asignación de direcciones IP planificando la duración de las concesiones. El problema es que si no se liberan direcciones, en un momento determinado no se podrá cumplir con nuevas solicitudes DHCP debido a que faltarán direcciones que puedan distribuirse.

En una red en la cual muchos equipos se conectan y desconectan permanentemente (redes de escuelas o de oficinas de ventas, por ejemplo), es aconsejable ofrecer concesiones por períodos cortos. En cambio, para una red compuesta principalmente por equipos fijos que se reinician rara vez, las concesiones por períodos largos son más que suficientes. No se olvide que DHCP trabaja principalmente por transmisión y que puede ocupar ancho de banda en redes pequeñas con alta demanda.

3.3.1.2. Aplicabilidad de DHCP a MANETs

DHCP permite la asignación automática de una dirección IP a un nodo por parte de un servidor DHCP. Un nodo que requiere una dirección IP contacta con un servidor DHCP y solicita una dirección. El servidor DHCP asignará la dirección dinámicamente a partir de un cierto conjunto de claves, y le cederá su uso al cliente. El cliente podrá usar a continuación esta dirección durante un cierto período de tiempo. Si el cliente desea mantener la dirección durante más tiempo, debe prolongar su reserva. Si el servidor DHCP no tiene un enlace directo con el cliente DHCP, es posible usar uno o más agentes de transmisión DHCP para enviar los mensajes a una subred diferente.

3.3.1.2.1. Suposiciones básicas de DHCP

DHCP trabaja con la suposición de que cada nodo en la MANET puede comunicarse directamente con el servidor DHCP o con un transmisor DHCP que pueda comunicar con el servidor o bien con otro transmisor.

La primera parte de la suposición es a menudo incorrecta en MANETs, puesto que cada router puede ver diferentes conjuntos de nodos MANET vecinos. Por otro lado, la segunda parte de esta suposición se fundamenta en la garantía de que la recurrencia acabará en alguna iteración. Dada la naturaleza dinámica de las MANEts no existe la garantía de que el servidor DHCP sea alcanzable, o de que no aparezcan ciclos a lo largo del camino.

Además, DHCP supone también que o bien existe un solo servidor DHCP en la red, o bien existen varios servidores DHCP repartidos a lo largo de la red, pero están

configurados manualmente de forma adecuada. Debido al dinamismo de los nodos MANET, no se puede asegurar en una red ad hoc que no se produzcan cambios topológicos que produzcan cambios topológicos que afecten a la configuración de parámetros de varios pudiendo crearse conflictos entre ellas (por ejemplo, no manejar conjuntos disjuntos de direcciones). Para evitar esto, los servidores necesitan una reconfiguración dinámica.

Igualmente, DHCP trabaja con la suposición de que deberían existir transmisores DHCP, que se benefician de una configuración manual. Debido al dinamismo de los nodos y la topología de las MANETs, esto no se puede garantizar. La configuración podría no ser correcta a lo largo del tiempo, por lo que necesitaría igualmente una reconfiguración dinámica.

3.3.1.2.2. Qué puede y qué no puede hacer DHCP en MANETs

DHCP como tal podría emplearse con alguna extensión para configurar direcciones. Sin embargo, la aplicabilidad de DHCP en este contexto es limitada. Si la topología es o llega a ser tal que un nodo no tiene acceso a un servidor DHCP, obien directamente o bien a través de un transmisor DHCP, DHCP deja de estar operativo.

DHCP como tal podría emplearse para mantener la unicidad de los prefijos y las direcciones. Sin embargo, la aplicabilidad de DHCP en este contexto es limitada. Desde el momento de que los distintos servidores DHCP no comprueban automáticamente el carácter disjunto de sus conjuntos de direcciones, si la topología es o llega a ser tal que la configuración de varios servidores DHCP entra en conflicto, no existe la garantía de que las direcciones que se asignen sean únicas.

3.3.2. SLAAC/NDP

Stateless Address Autoconfiguration (SLAAC) permite la configuración automática de una dirección IP sin la necesidad de conectarse a ningún servidor. El nodo construye primero una dirección Ipv6 provisional para asignar a su identificador (en la mayoría de los casos la dirección MAC) al prefijo del enlace local. Después se realiza un proceso de detección de dirección duplicada que verifica que no existe enlace con ningún nodo con la misma dirección inundando la red con mensajes NDP (Neighbor Discovery Protocol). Si la dirección no es única el proceso de autoconfiguración abortará. Tras realizar un test de unicidad satisfactorio, el nodo debe solicitar un prefijo de algún router con el que exista un enlace mediante el envío de mensajes NDP. Asignará de nuevo su identificador al prefijo del router y repetirá la secuencia del proceso de test de unicidad.

3.3.2.1. Aplicabilidad de SLAAC/NDP a MANETs

3.3.2.1.1. Suposiciones básicas de SLAAC/NDP

SLAAC se basa en la señalización de NDP, que trabaja con la suposición de que cada nodo de la MANET puede comunicarse directamente con todos lo demás nodos de la MANET, por ejemplo si todos están conectados por un enlace multicast. Esta

suposición es normalmente errónea en una MANET, puesto que cada nodo tiene un conjunto diferente de vecinos MANET.

3.3.2.1.2. Qué puede y qué no puede hacer SLAAC/NDP en MANETs

SLAAC podría ser empleado con alguna extensión para configurar las direcciones y mantener su unicidad cuando, por ejemplo, no existe ningún servidor DHCP disponible. Sin embargo, la aplicación de SLAAC en este contexto es limitada. Puesto que los mensajes NDP no son transmitidos más allá del enlace (en términos de MANETs no más allá del primer salto). Si la topología es tal o llega a ser tal que, que la MANET no está contenido en un enlace simple, no existe garantía de que las direcciones asignadas sean únicas, puesto que la señalización no alcanzará todos los nodos implicados.

3.3.3. DHCP - PD

DHCP – PD es una versión de DHCP que permite la asignación automática de prefijos IPv6 a routers empleando IPv6 usando DHCP. Un router debe solicitar la asignación de un prefijo por parte de un servidor DHCP enviando una solicitud DHCP, incluyendo la opción Prefix Delegation. El servidor DHCP debe entregar un subprefijo al router (por ejemplo, un subconjunto de su conjunto de direcciones). El mensaje DHCP que contiene la opción de Prefix Delegation debe ser transmitido a través de uno o más transmisores DHCP.

3.3.3.1. Aplicabilidad de DHCP – PD a MANETs

3.3.3.1.1. Suposiciones básicas de DHCP – PD

DHCP - PD está basado en DHCP y, por tanto, tiene en cuenta las mismas suposiciones de alcanzabilidad de los servidores, y reconfiguración dinámica de de los servidores y transmisores.

3.3.3.1.2. Qué puede hacer y qué no puede hacer DHCP – PD en MANETs

DHCP – PD podría ser usado con alguna extensión para asignar prefijos y para mantener la unicidad. Sin embargo, la aplicabilidad de DHCP – PD en este contexto es limitada. Si la topología es o llega a ser tal que el router MANET no puede comunicarse con servidor DHCP, DHCP – PD deja de estar operativo. Además, si la topología es o llega a ser tal que entran en conflicto de configuración formando parte de la misma MANET, no hay mecanismos de reconfiguración automáticos disponibles que permiten a los servidores adaptarse dinámicamente a la situación.

3.4. Requisitos que debe cumplir la solución

Esta es una lista de los requisitos que debería cumplir una posible solución para la autoconfiguración de direcciones IPv6 en MANETs propuesta por el IETF:

- Las soluciones deben configurar los interfaces MANET con direcciones IPv6 que sean únicas en la MANET.

- Las soluciones deben configurar los routers de la misma MANET con prefijos disjuntos.

- La solución debe trabajar de forma independiente al protocolo de enrutamiento de la MANET. Sin embargo, debe tener en cuenta la existencia de un protocolo de enrutamiento con fines de optimización.

- Las soluciones deben proporcionar un mecanismo para prevenir y lidiar con conflictos de direcciones o prefijos (debido a la fusión de redes, cambios en los miembros de la red, preconfiguración o falta de configuración).

- Las soluciones deben ser diseñadas teniendo en cuenta las características particulares de las MANETs, incluidas su naturaleza multi-hop y la potencial asimetría de los enlaces.

- Las soluciones deben cumplir sus objetivos con una baja sobrecarga del control

- Las soluciones deben cumplir sus objetivos con un bajo retardo.

- Las soluciones deben presentar compatibilidad hacia atrás con otros standard definidos por el IETF.

- Las soluciones no deben requerir modificaciones de los protocolos existentes en los interfaces no MANET y en los routers no MANET.

- Las soluciones deben tener en cuenta las amenazas de seguridad de las direcciones consideradas en los mecanismos existentes de autoconfiguración para IPv6. Además deberán tener en cuenta las posibles amenazas específicas de las MANETs.

- Las soluciones deberán funcionar en MANETs conectadas a una red externa por medio de varios routers, así como en MANETs conectadas a múltiples redes externas.

- En el caso de las MANETs subordinadas, las soluciones deberán tener un impacto mínimo en el sistema de enrutamiento de la red o redes externas a las que la MANET está conectada. En particular, esto incluye lo siguiente:

    - Las soluciones no deben impedir la agrupación de los prefijos en el borde de la MANET subordinada.

- Las soluciones deben soportar la transición de un tipo de escenario MANET a otro (por ejemplo, de un escenario subordinado a uno autónomo o viceversa).

- Las soluciones deben ser diseñadas de forma modular, cada módulo direcciona según un subconjunto específico de requisitos o escenarios.

3.5. Soluciones para MANETs

Como todos los nodos IP, los nodos MANET necesitan una dirección IP configurada en su interfaz de red, válida dentro de la MANET para las comunicaciones intra-MANET, y válida globalmente para comunicarse con dispositivos en Internet. SLAAC, NDP, DHCP y DHCP-PD proporcionan únicamente soluciones parciales con respecto a los objetivos enumerados en la sección 3.2. Como señalamos en las secciones 3.3.1, 3.3.2 y 3.3.3, estos protocolos tal cual son no pueden tratar con la naturaleza dinámica, multi-hop y distribuida de las MANETs. Por lo tanto, se están centrando nuevos esfuerzos en desarrollar protocolos de configuración específicos para MANET, que si bien todavía no ha logrado una solución completamente satisfactoria, han aportado mejoras importantes. Vamos a comentar alguno de ellos.

3.5.1. Adress Reservation and Optimistic Duplicated (AROD)

AROD propone un esquema de autoconfiguración de direcciones distribuido para MANETs usando un mecanismo de reserva de direcciones en combinación con un mecanismo optimista de detección de direcciones duplicadas (Duplicated Address Detection). De este modo, los nodos que existen en la MANET tienen a su lado nodos con una dirección propia y una dirección reservada. Un nuevo nodo uniéndose a la MANET selecciona un nodo agente y solicita la dirección reservada del agente. Si el agente tiene una dirección reservada se la da al nuevo nodo. En otro caso el agente busca entre sus vecinos algún nodo con una dirección reservada. Después de la asignación de una dirección correcta, el agente genera dos direcciones aleatorias, comprueba su unicidad realizando un DAD, da una dirección al nuevo nodo como dirección reservada y mantiene otra para sí mismo. El proceso de DAD es considerado como optimista puesto que sólo es realizado una vez, sea satisfactorio o no. Se espera que el mecanismo AROD de asignación de direcciones tenga una latencia baja debido a la rápida adquisición de la dirección reservada y a la baja sobrecarga de las comunicaciones gracias al optimista comportamiento del DAD. El mecanismo es válido para IPv4 e IPv6. Sin embargo, sólo existe una visión general del algoritmo y no una especificación detallada del protocolo. Además, el mecanismo no considera la fusión de MANETs.

3.5.2. Ad Hoc IP Adress Autoconfiguration

La clave de este esquema es manejar las particiones de la MANET y las fusions, resolviendo direcciones duplicadas. El mecanismo está diseñado para IPv4 e IPv6 y los tipos de mensajes para los procesos están ya especificados. Se genera una dirección aleatoria para el ámbito MANET para IPv4 e IPv6, determinando su unicidad en la MANET, y manteniendo la corrección chequeando permanentemente las direcciones para detectar y solucionar los conflictos causados por la fusión de MANETs. Duplicate Address Detection (DAD) está basado en un mecanismo híbrido, aplicando Strong DAD para determinar la duplicación de una dirección en una partición específica y Weak DAD para localizar direcciones duplicadas después de que algunas particiones se hayan fusionado. De esta forma, el mecanismo Strong DAD está basado en un protocolo

request-reponse donde un nodo que intenta asignar una nueva dirección a uno de sus interfaces envía un mensaje Address Request (AREQ) a través de la MANET y espera a recibir un mensaje Address Reply (AREP). Cuando un nodo no recibe ningún AREP considera que su dirección es única. Se pueden emplear retransmisiones, pero es necesario tener cuidado con los contadores para evitar una sobrecarga en la red. En el mecanismo Weak DAD se emplea una combinación de direcciones IP y una clave para determinar las direcciones duplicadas. Weak DAD requiere un campo adicional "Key" en las tablas de rutas, lo que obliga a rediseñar las tablas de rutas existentes. El valor key es asignado a cada interfaz de red. Es añadido a los paquetes de control del protocolo de enrutamiento, como los paquetes route Discovery, los paquetes HELLO, etc., y los nodos intermediarios deben mantener el valor key de cada dirección en la tabla de rutas. Se detecta un conflicto de direcciones en caso de que se reciba un paquete de control con la extensión Interface-Key con una dirección IP que coincida con una existente en la tabla de rutas con un valores del campo key diferentes.

3.5.3. Extensible MANET Auto-configuration Protocol (EMAP)

EMAP es un protocolo de autoconfiguración para MANETs con direccionamiento IPv4 o IPv6. EN MANETs aisladas, EMAP puede ser usado para autoconfigurar una dirección IP única (o al menos con muchas posibilidades de ser única) en el ámbito de la MANET. En MANETs híbridas (con conexión a Internet) puede ser usado incluso para autoconfigurar direcciones IP globalmente direccionables. Además, EMAP está diseñado para descubrimientos de servicios para MANETs.

Un punto clave de EMAP es el soporte de proxies, por ejemplo, un nodo MANET que no proporciona un servicio concreto pero contesta a las solicitudes en nombre de los nodos que proporcionan los servicios.

Han sido especificados dos tipos de mensajes EMAP, los mensajes EMAP_REQUEST y los mensajes EMAP_REPLY. Al lado del campo de tipo, estos menajes contienen un campo de código que señala la función del mensaje. Los mensajes pueden ser de tipo Duplicate Address Detection (DAD), Global Configuration (GC) y DNS Server Discovery (DS). Dependiendo de su tipo y funcionalidad, EMAP o bien envía sus mensajes mediante flooding o bien mediante enrutamiento unicast. No está especificado como realizar el flooding, pero es recomendable utilizar mecanismos de optimización.

Respecto a la autoconfiguración de una dirección IP local de una MANET (denotada como configuración local), un nodo selecciona una dirección temporal empleada como Originator Address in los mensajes Request DAD y una dirección IP tentativa que es usada como Requested Address en los mensajes Request DAD y como Originator Address en los mensajes Reply DAD.

Para IPv4, ambas direcciones son tomadas de la subred 169.254/16 con los 16 bits menos significativos seleccionados de manera aleatoria en el rango 1 – 2047 para la dirección temporal, y en el rango 2048 – 65534 para la dirección tentativa. Para IPv6 no se ha tomado ninguna decisión aún. Asimismo, se forman Unique Local Address (direcciones locales únicas) con un campo especial Global ID reservado para MANETs,

el ID de la subred elegido igual a los 16 bits menos significativos en IPv4, y la dirección del interfaz EUI-64 del interfaz solicitante usado como identificador de interfaz. DAD genera la dirección tentativa de forma similar a Concerning Global Configuration (GC). Para la autoconfiguración de una direcciónglobal válida que permite a los nodos MANET comunicarse con nodos en Internet, un Internet Gateway (IGW) puede inundar la red periódicamente con mensajes GC_REP o enviarlos de manera unicast en respuesta a un mensaje GC_REQ. En el GC_REP va ímplicito el prefijo global puesto que el destinatario lo calcula a partir de la dirección de procedencia del mensaje y de la longitud del prefijo.

3.6. Detección de Gateways

Como hemos visto anteriormente, las MANETs pueden ser autónomas o bien subordinadas, estando unidas a una o más redes externas. Esta red externa puede ser Internet, a la que estará conectada mediante una o varias Internet Gateways. Las Internet Gateways pueden ser fijas o móviles, simples o múltiples, equipadas con interfaces cableadas y/o inalámbricas. Para poder conectarse con nodos pertenecientes a Internet, un nodo MANET debe ser capaz realizar una detección de Gateways para obtener información acerca deque nodo puede enviar paquetes destinados a Internet. Además, las MANETs deben considerar una extensión del concepto de dirección, puesto que deben conocer que direcciones pueden ser encontradas dentro de la red y cuáles deben alcanzarse a través de Internet.

Los protocolos de encaminamiento diseñados para redes ad-hoc aisladas, no permiten el descubrimiento de gateways. Estos protocolos no son capaces de transmitir paquetes desde la red ad-hoc hacia la red IP fija. Para que la comunicación sea posible, se modifican los protocolos de encaminamiento en redes ad-hoc para poder descubrir gateways y hacer realidad la posibilidad de comunicar una red ad-hoc con la red IP fija.

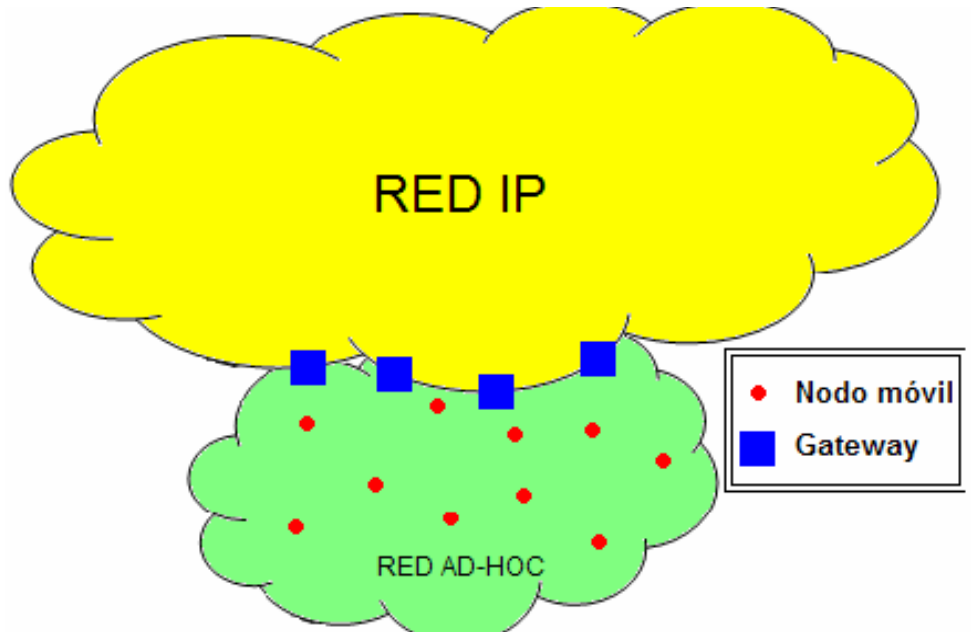


Figura 3.1. *Interconexión mediante Gatewavs.*

Modificando los protocolos de encaminamiento de redes ad-hoc queremos que sean capaces de establecer y mantener rutas hacia los gateways. Estos protocolos modificados se encargan de hacer la conversión de la pila de protocolos de la red ad-hoc a la pila de protocolos de la red fija (Figura 3.2).

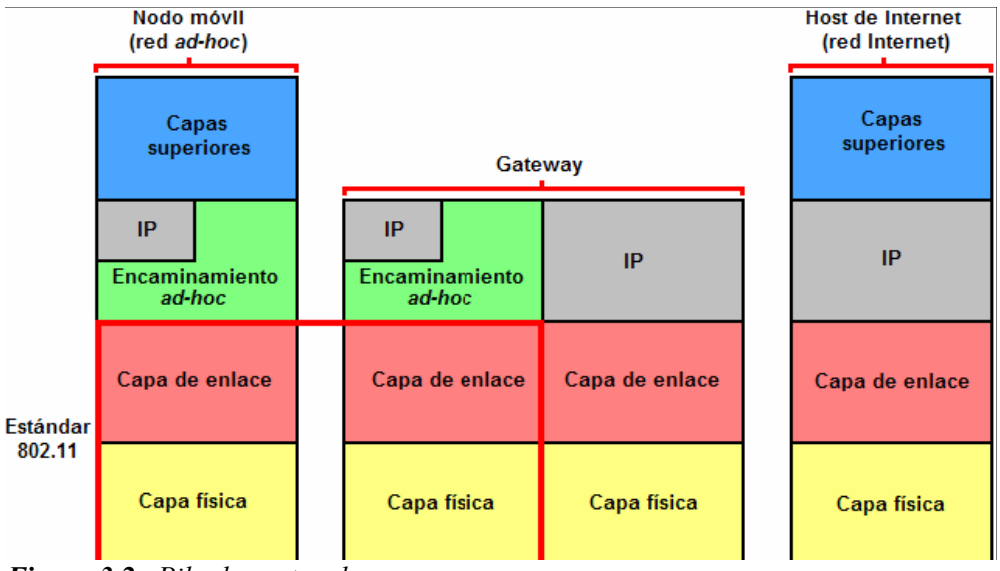


Figura 3.2 . *Pila de protocolos.*

3.6.1. Tipos de protocolos de detección de Gateways

En estos días se trabaja con tres protocolos de descubrimiento, los cuales han sido modificaciones de protocolos de encaminamiento en redes ad-hoc. Estos protocolos de encaminamiento en redes ad-hoc han sido extendidos para poder trabajar a nivel IP y poder, por lo tanto, interactuar con nodos móviles y gateways al mismo tiempo. Existen tres protocolos de enrutamiento, y por lo tanto, los mismos que de descubrimiento: proactivos, reactivos e híbridos.

3.6.1.1. Proactivos

Cuando se habla de protocolos de descubrimiento de gateways pro-activos, su principal característica es que es el gateway quién indica a los nodos móviles cual es su disponibilidad, iniciando el envío de mensajes. El gateway hace un envío broadcast cada cierto tiempo, un tiempo no aleatorio para no hacer envíos innecesarios. Cuando un nodo móvil recibe el paquete, actualiza su tabla de rutas si ya tiene esa ruta creada o la añade si este gateway no existía en su tabla. Seguidamente el nodo reenviará este anuncio del gateway en modo broadcast, para que los nodos móviles dentro de su rango de cobertura sepan esa ubicación. Esto puede provocar envíos no necesarios, además de ocupar ancho de banda y tiempo en actualizar todos los nodos de la red ad-hoc.

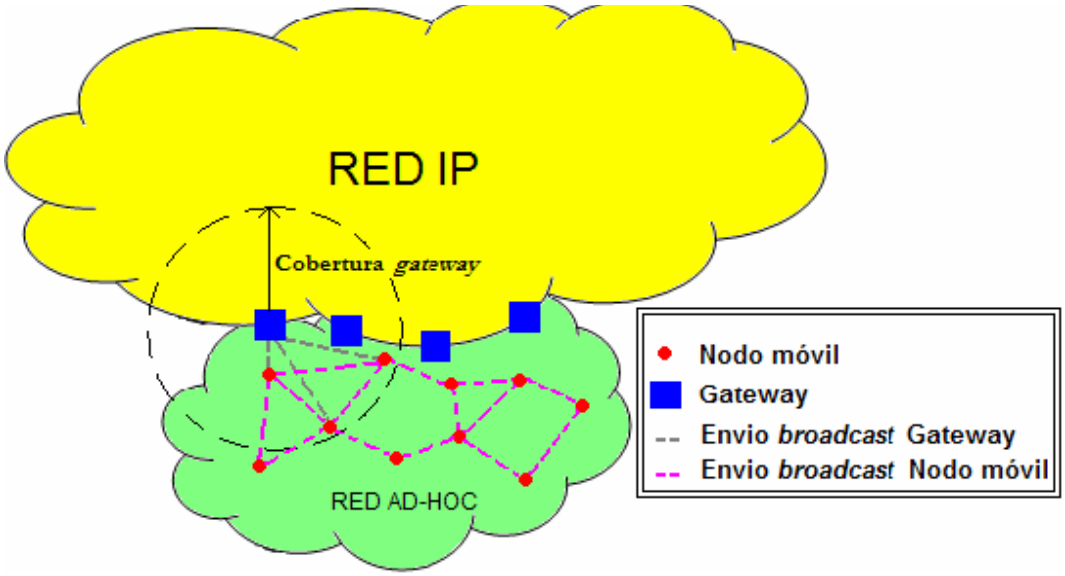


Figura3.3 . *Actualización y creación de rutas en modo proactivo.*

3.6.1.2. Reactivos

Este tipo de protocolo de descubrimiento de gateways se caracteriza porque es el nodo móvil de una red ad-hoc quien hace el envío de paquetes para saber la disponibilidad de los gateways. Cada nodo envía un mensaje de pregunta para saber que gateways existen para poder crear o actualizar rutas. Cuando un gateway recibe un paquete de pregunta de un nodo móvil, contesta al nodo que envió la petición indicando su dirección IP. Si un nodo móvil recibe un mensaje de otro nodo o gateway, lo procesará y lo reenviará otra vez, actualizando su tabla. Como gran desventaja nos encontramos a los nodos cercanos a un gateway, puesto que procesarán y reenviarán muchos mensajes de otros nodos. Aunque también pasa con otros protocolos de descubrimiento de gateways.

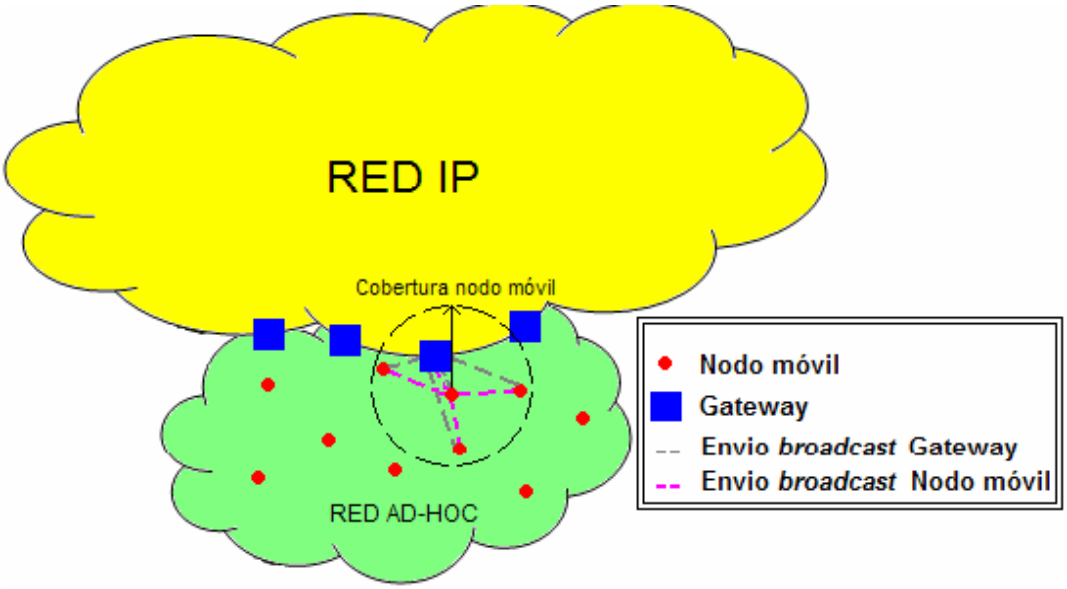


Figura 3.4 . *Actualización y creación de rutas en modo reactivo.*

3.6.1.3. Híbridos

Es la combinación entre pro-activos y reactivos, cogiendo las ventajas de cada uno de ellos e intentado minimizar las desventajas. Separan la red ad-hoc en diferentes regiones, usando las características de los proactivos en nodos cercanos y reactivos en los nodos alejados con respecto al gateway. El gateway enviará mensajes hasta un cierto alcance de la red ad-hoc, y cada nodo de la red móvil hará una búsqueda cuando necesite obtener una ruta para encaminar paquetes hacia la red fija y no le haya llegado ningún aviso acerca de la disponibilidad dentro de una determinada zona. El gateway enviará mensajes indicando su disponibilidad dentro de una determinada zona en la red ad-hoc determinada por un número de saltos desde el gateway. A partir de ese último salto se separará la parte pro-activa y reactiva del envío de mensajes por parte del gateway y los nodos móviles.

3.6.1.4. Detección de Gateways con EMAP

En el protocolo EMAP, una Internet Gateway envía o bien periódicamente o tras una solicitud un mensaje de presentación a la MANET, especificando un prefijo global (calculado a paritr de la dirección del generador y de la longitud del prefijo) y la

dirección del Gateway. En el caso de que un nodo reciba una presentación de una Internet Gateway de varias Gateways, un nodo MANET debe seleccionar una como router por defecto para conectarse a Internet. La selección se debe basar en el campo de distancia en los mensajes GC_REP, el cual se incrementa en cada nodo MANET antes de reenviarlo.

3.6.2. Conectividad global para MANETs basadas en IPv6

Es un mecanismo que permite a la MANET adquirir información de enrutamiento global de un Internet Gateway y como comunicarse a través del Gateway con otros nodos de Internet. Un nodo MANET descubre un Internet Gateway recibiendo un aviso de Internet Gateway. Los avisos de Internet Gateways pueden ser emitidos periódicamente o en respuesta a una solicitud. De la recepción de una advertencia de Internet Gateway un nodo aprende un prefijo, una dirección IP válida usando el prefijo recibido en combinación con su dirección del interfaz EUI-64, e introduce la dirección del Gateway como acceso a Internet en su tabla de rutas.

Los mensajes de anuncio y de solicitud puede ser implementados como modificaciones de los mensajes Router Advertisement y Router Solicitation del Neighbor Discovery Protocol (NDP) de IPv6, o como un mensajes de control adicionales del protocolo de enrutamiento usado. Esto es un requisito crítico, puesto que necesitará una modificación especial de IPv6 o del protocolo de enrutamiento MANET concreto. Los creados especifican una modificación de los mensajes NDP pero no mensajes adicionales del protocolo de enrutamiento. La modificación de NDP prevé un flag Advertisement/Solicitation que señala si un mensajes puede ser enviado a través de múltiples de saltos, o se reduce su ámbito a un solo salto (caso de los mensajes NDP ordinarios).

3.6.3. Funcionalidad de detección de Gateways en protocolos de enrutmaiento MANET

Algunos protocolos de enrutamiento MANET especifican detección de Gateways, por ejemplo DYMO y OLSR. En el IETF RFC 3626 se especifica como parte del Optimized Link State Routing Protocol (OLSR) un mecanismo que soporta detección de Gateways mediante el uso de mensajes Host and Network Association (HNA). Un MANET Gateway, un nodo que tiene un interfaz para acceder a la MANET (mediante OLSR) e interfaces no MANET, envía periódicamente mensajes a la MANET, conteniendo información de prefijos de redes externas alcanzables a través de la MANET. La información de prefijo puede dar la ruta por defecto para acceder a Internet. De la información contenida en los mensajes HNA, los nodos receptores aprenden las rutas a las redes externas, por ejemplo, la dirección del Gateway que da acceso a Internet. Sólo se ha especificado el formato de los paquetes IPv4 para los mensajes HNA. Sin embargo, hay implementaciones que ya proporcionan el procesado de los mensajes HNA bajo IPv6. En DYMO, un nodo Gateway que está conectado a Internet responde a un mensaje Route Request (RREQ) con una dirección de destino exterior con un mensaje Route Reply (RREP). De esta forma, el nodo Gateway activaría el bit G en cualquier Route Element (RE) enviado o procesado ya el bit G indica a los

nodos de la MANET que el RBNodeAddress está conectado a Internet y es capaz de dirigir paquetes de datos a todos nodos exteriores fuera de la subred MANET.

3.7. Descubrimiento de servicios

Habiendo obtenido la dirección IP automáticamente y detectado un Gateway a Internet, un nodo MANET puede estar interesado en autoconfigurar servicios adicionales, como las direcciones IP de los servidores DNS. Qué servicio se requiere y, por tanto, qúe servicio es necesario buscar depende del escenario y de la aplicación.

3.7.1. Descubrimiento de servicios en EMAP

Como mencionamos anteriormente EMAP es un protocolo de autoconfiguración para redes MANETs basadas en IPv4 o IPv6 que soportan descubrimiento de servicios. En el draft actual sólo se especifica el descubrimiento de servidores DNS, pero debido a su extensibilidad el protocolo puede ser mejorado con soporte para descubrimiento de otros servicios. Para descubrir servicios, los nodos que conocen la dirección de las entidades que proporcionan estos servicios (por ejemplo, un servidor DNS) comunican su información a la MANET o bien, de forma periódica, o bien en respuesta a una solicitud explícita del cliente. El nodo comunicante puede proporcionar el servicio él mismo o conocer la dirección del servidor respectivo, actuando como una especie de Proxy de autoconfiguración.

Para el descubrimiento y configuración de servidores DNS (DS), una entidad llamada DNS Server Advertiser (DSA) proporciona la dirección IP de un servidor DNS primario y, tal vez, la de un servidor secundario. El DSA debe coubicarse con un Internet Gateway (IGW). De forma similar a la detección de Gateway, un DSA puede inundar con un DS_REP a la MANET de forma periódica y enviar un DS_REP en respuesta a un mensaje DS_REQ de un cliente. Los mensajes DS_REP contienen la dirección IP (IPv4 y/o IPv6) de un servidor DNS primario y uno secundario.

3.8. Implicaciones de seguridad

Una cuestión importante de seguridad es el mantenimiento de la confidencialidad y la integridad de la información transmitida entre dos nodos de la MANET. Esto es equivalente a proporcionar seguridad extremo a extremo en otros tipos de redes y las técnicas existentes son, por tanto, aplicables.

Una cuestión ortogonal respecto a la seguridad de los protocolos MANET es asegurar la integridad de la red. Así que, los protocolos MANET en general, permiten a cualquier nodo participar en la red (bajo la suposición de que todos los nodos son bienintencionados). Esta suposición es incorrecta si existe la posibilidad de la presencia de nodos malintencionados (algo común en las MANETs) provocará un fallo en la integridad de la red. Los comportamientos maliciosos incluyen, Hamming (que provoca DoS), generación de tráfico incorrecto, transmisión de tráfico incorrecto, etc.

El ataque DoS puede penalizar laa operación de las soluciondes de autoconfiguración IP, a través del incremento de la sobrecarga de señalización y alterar,

por tanto, el tiempo de convergencia. El ataque man-in-the-middle puede causar intercepciones de mensajes de autoconfiguración IP y, por tanto, el fallo de la operación, modificación de los mensajes por ejemplo, las direcciones o prefijos asignados durante su transferencia, causando conflictos de direcciones o prefijos, suplantación, al hacerse pasar un nodo por otros perteneciente a la MANET.

La mayoría de las soluciones existents protegen la integridad de la red a través de la garantía de autenticación a partir de una autoridad de certificación confiable, que proporciona las claves que permiten realizar la autenticación. Sin embargo, en el caso de los escenarios autónomos MANET, puede no haber una autoridad central, o puede no ser confiable a priori por los nodos d ela MANET.

La encriptación es el mecanismo es el más esencial de los mecanismos de seguridad existentes. Sin embargo, puede afectar a los tiempos de convergencia o suponer un coste operacional demasiado elevado en el contexto de las MANET, ya que una parte significativa de los nodos participantes en la MANET pueden tener recursos limitados.

Otra cuestión específica de las MANETs está relacionada con el egoísmo de ciertos nodos, conocido como el problema del nodo egoísta. Este comportamiento puede causar la no cooperación entre nodos MANETs durante la autoconfiguración IP y, por tanto, afectar al correcto funcionamiento de los mecanismos de autoconfiguración.

4. Control de Acceso al Medio (MAC)

Las redes inalámbricas se caracterizan por utilizar el canal radio como medio de transmisión. Este medio es compartido por el conjunto de usuarios de la red inalámbrica. Al tratarse de un medio compartido, es imprescindible establecer una serie de normas o leyes que rijan de una manera eficiente y ordenada el acceso a dicho medio. Este conjunto de normas son los llamados protocolos de acceso al medio, protocolos MAC, del inglés, *Medium Access Control protocols*. El propósito de los protocolos MAC en entornos inalámbricos es el de gestionar de una manera eficiente los escasos recursos radio disponibles. Su principal objetivo es el de optimizar el uso del canal radio, ya que el espectro de frecuencia disponible es limitado y el número de usuarios crece día a día, y no sólo el número de usuarios, si no que también los requerimientos de dichos usuarios. La creciente demanda de información multimedia, la combinación de voz y datos, el envío de video y audio, etc., exige nuevos y más estrictos requisitos en términos de grandes anchos de banda necesarios para cada usuario y, en muchos casos, con exigencia de calidad asegurada (QoS, del inglés, *Quality of Service*). Todo ello lleva a la necesidad de diseñar y proponer nuevos protocolos de acceso capaces de responder a las exigencias actuales.

Para el caso concreto de las redes ad hoc, la ausencia de infraestructura previa capaz de gestionar y controlar el acceso al canal radio, hace imprescindible la cooperación entre todos los usuarios para poder llevar a cabo un uso eficiente de los recursos. Sin duda, esta falta de centralización no es más que una dificultad añadida y un nuevo reto para el diseño de protocolos MAC eficientes. Este problema puede abordarse de diferentes maneras que describimos a continuación:

- Reparto estático del canal. Es la técnica tradicional para asignar un canal único a múltiples estaciones. Hay dos formas básicas:

    - Multiplexación por División en Frecuencias (FDM, Frequency-Division Multiplexing)
    - Multiplexación por División del Tiempo (TDM, Time-Division Multiplexing)

  Con ambas aproximaciones, para n estaciones el canal se divide en n partes iguales y a cada estación se le asigna una parte. En el caso de FDM se divide el ancho de banda y a cada estación le corresponde un rango de frecuencias, en el caso de TDM el tiempo disponible se divide en ranuras. Este esquema es adecuado para estaciones que transmiten de forma continua pero es muy poco eficiente si el número de estaciones activas en cada momento es bajo, o si el tráfico se produce a ráfagas, ya que si una estación permanece en silencio, se desperdician los recursos asignados a ella.

- Reparto dinámico del canal. La asignación del canal se hace teniendo en cuenta las necesidades de las estaciones en cada momento. Dentro de este enfoque se incluyen tres categorías:

    - Protocolos de contienda. Cada estación emite cuando tiene datos para emitir, y por tanto puede haber conflictos al usar el canal (colisiones). A

estos protocolos también se les llama de contención o de competición (contention protocols). En ocasiones también se les denomina de acceso aleatorio (random protocols), puesto que el patrón que van a emplear las estaciones para intentar ocupar el medio es impredecible (desde el punto de vista de la capa de acceso al medio). Entre estos se encuentran ALOHA y todas sus variantes, CSMA, CSMA/CD. Los protocolos de contienda son los más apropiados en condiciones de carga baja en el canal, por su bajo retardo. Con el aumento del tráfico aumenta el consumo de recursos destinados al arbitraje del canal.

- Protocolos libres de colisiones. En la actualidad no están demasiado extendidos. Pueden subdividirse a su vez en dos clases:

  - Basados en reserva. A cada estación se le asigna una ranura de tiempo en la que debe anunciar si desea enviar alguna trama.
  - Basados en consulta o sondeo. Normalmente emplean un testigo (token) que circula por el medio, para que una estación pueda transmitir debe capturar el testigo. Están indicados para entornos donde el canal está muy ocupado, o si se requieren garantías de posibilidad de envío. Si hay poco tráfico, el mecanismo que evita la colisión resulta una carga no necesaria.

- Protocolos de contienda limitada. Representan una aproximación mixta entre los protocolos de contienda y los protocolos libres de colisiones. Dividen las estaciones en grupos y usan algoritmos libres de colisiones para establecer qué grupo de estaciones puede ocupar el medio en cada momento. Las estaciones pertenecientes a un mismo grupo compiten entre sí empleando algoritmos de contienda. Si la carga es baja, será apropiado usar pocos grupos, pudiendo haber muchas estaciones en cada uno. En el caso extremo habría un único grupo, con lo que el algoritmo degeneraría en un protocolo de contienda. Si la carga es alta los mejores resultados se obtendrán con muchos grupos, con pocas estaciones en cada uno: llevando este enfoque al límite tendríamos tantos grupos como estaciones, resultando un protocolo libre de colisiones.

Una vez hemos visto los diferentes tipos de protocolos, vamos a ver las diferentes tecnologías de acceso al medio:

Debido al escaso ancho de banda disponible in MANETs, el diseño de protocolos de control de acceso al medio (MAC) eficientes y efectivos que regule el acceso al medio compartido es sujeto de numerosos estudios en los últimos años. Se han propuesto muchos protocolos MAC para mitigar los efectos adversos de los terminales ocultos a través de la prevención de colisiones. La mayoría de los esquemas de prevención de colisiones como el "Carrier Sense Multiple Access with Collision Avoidance" (CSMA/CA) dentro de los protocolos MAC más populares, el protocolo MAC 802.11, son iniciados por el emisor, incluido el intercambio de pequeños paquetes request-to-sent (RTS) y clear-to-send (CTS) entre un par de nodos antes de la transmisión del paquete de datos o del opcional paquete acknowledgment.

4.1. Problemas de la capa MAC

Como hemos comentado anteriormente, el medio inalámbrico es un medio compartido, por lo que los nodos cercanos deberán competir para acceder a él. En el caso de que no exista una entidad o bien central o bien distribuida que gestione el acceso al medio inalámbrico, se producirán colisiones entre los paquetes enviados, lo que provocará la pérdida de información. Para evitar esto es necesaria la existencia de un protocolo que asigne a cada nodo una fracción de tiempo para poder emitir. Tradicionalmente, se han empleado algoritmos basados en "Carrier sense", es decir, un nodo antes de emitir escuchaba el medio. Si no escuchaba ninguna emisión, empezaba a emitir. En caso contrario, esperaba a que el medio quedase libre. Este algoritmo presenta dos problemas fundamentales, que serán descritos a continuación y son dos de las principales preocupaciones de los investigadores que trabajan con protocolos de acceso al medio, el terminal oculto y el terminal expuesto.

4.1.1. El problema del terminal oculto

En Carrier Sense Multiple Access (CSMA), todos los terminales escuchan el canal al que se encuentran conectados. No se inicia la transmisión de ningún paquete si se detecta que el canal está ocupado. Esto requiere que todos los terminales móviles puedan recibir todas las señales de los nodos que emplean el mismo el mismo canal. Sin embargo, esto no se puede garantizar en las MANETs. Esto se puede comprobar con un ejemplo muy sencillo, como se muestra en la Figura 3.5.

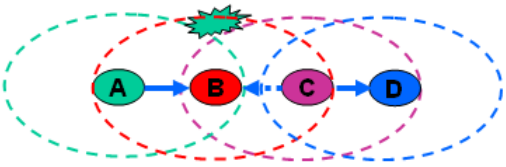


Figura 3.5. *Problema del terminal oculto.*

Existe un nodo B que es alcanzable desde A y C. Sin embargo, A y C no se pueden ver entre ellos. A está transmitiendo a B. C quiere transmitir a B. Escucha el canal y no detecta que esté ocupado, puesto que la señal de A no llega hasta C. Por lo tanto, C comienza a emitir, provocando colisiones entre los paquetes transmitidos por A y los suyos propios. Esto conllevará pérdida de información y la retransmisión de la información, lo que producirá a su vez, una degradación de la productividad.

4.1.2. El problema del terminal expuesto

En este caso, la escucha del medio provoca una pérdida de la productividad al considerar el medio ocupado en situaciones que no provocan en realidad colisiones. Se produce cuando un nodo que quiere transmitir un paquete escucha el medio y detecta que otro nodo está emitiendo, pero ambas emisiones tienen destinatarios diferentes y, en

ningún caso alcanzan ambos nodos provocando colisiones. Vamos a explicarlo mejor con el ejemplo mostrado en la Figura 3.6.

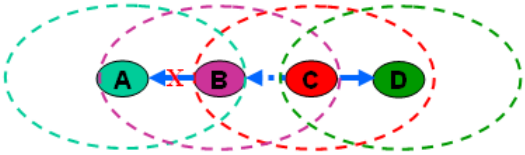


Figura 3.6. *Problema del terminal expuesto.*

C quiere comunicarse con B y D. B quiere comunicarse con A y C. Cuando C comienza a emitir, B detecta el medio como ocupado y espera hasta que el medio quede libre. Sin embargo, vemos que podría emitir perfectamente, puesto que su transmisión no produce colisiones ni A ni en C.

4.2. Protocolos MAC

Son muchos los protocolos de acceso que se han propuesto hasta la fecha, aunque no todos han sido utilizados en sistemas comerciales de telecomunicaciones. La mayoría de estos protocolos de acceso se han desarrollado, principalmente, para entornos centralizados donde tan sólo hay un receptor y muchos transmisores. Este clásico modelo de sistema no encaja con el modelo de redes distribuidas ad hoc, donde las redes se deben configurar dinámicamente y adaptarse a los cambios de la topología de red.

Los primeros y más sencillos protocolos MAC fueron los protocolos ALOHA y Slotted-ALOHA, que han sido ampliamente utilizados como protocolos de acceso aleatorio a pesar de su baja eficiencia (18% y 36% de utilización máxima de canal, respectivamente) y su poca estabilidad en condiciones de elevada carga ofrecida de tráfico. Los inconvenientes presentados por los protocolos ALOHA y S-ALOHA dieron pie a los llamados algoritmos de resolución de colisiones, también llamados CRA (*Collision Resolution Algorithm*). Algunos de estos, llamados algoritmos en árbol alcanzan eficiencias del 56%, mientras que otros, mediante el uso de ranuras temporales reservadas a la petición de acceso alcanzan eficiencias muy superiores, como por ejemplo el *Arrival Random Access Protocol* (AARA), que alcanza valores del 86% usando tres ranuras de acceso por cada slot de datos. El *Distributed Queue Request Update Multiple Access* (DQRUMA) es otro de los protocolos ampliamente utilizados que utiliza ranuras de acceso. La principal característica de este protocolo es que todo el control de acceso queda centralizado en la estación base, facilitando así la modificación sobre la marcha de los criterios de asignación de recursos de transmisión de los terminales en función de las necesidades del sistema. También destacan los protocolos basados en escuchas del canal radio *Carrier Sense Multiple Access* (CSMA) con sus variantes de detección y evasión de colisiones, que se utilizan en algunos estándares de comunicaciones como el IEEE 802.11 (Ethernet). Los sistemas de paso de testigo (*token ring*) también han sido ampliamente utilizados ya que evitan por completo la existencia de colisiones. Para el caso de los sistemas de comunicaciones móviles se han desarrollado algunos protocolos que tratan de realizar tareas equivalentes a la detección del estado del canal que realizan los pensados para redes fijas. Entre este grupo de protocolos destaca el *Inhibit Sense Multiple Access* (ISMA).

Como ya se ha comentado, la mayoría de estos protocolos se han desarrollado para entornos centralizados, y hasta el momento, son pocos los esfuerzos dedicados al desarrollo de protocolos para redes distribuidas. En la actualidad, la norma IEEE 802.11b especifica el protocolo MACA como protocolo de acceso en modo AD HOC.
En este sentido, el trabajo realizado en este proyecto supone una aportación a la comunidad científica, y por qué no, a la comunidad empresarial y comercial, proponiendo un novedoso protocolo de acceso diseñado para redes distribuidas ad hoc.

4.2.1. CSMA/CA

Carrier Sense Multiple Access with Collision Avoidance o CSMA/CA es un protocolo que sensa el canal antes de producir una transmisión, y si éste está ocupado utiliza un algoritmo de backoff para volver a sensar el canal hasta encontrarlo libre. Una vez que el canal está libre, resuelve los problemas mencionados haciendo handshaking de señales RTS (Request To Send) CTS (Clear To Send) DATA (Datos). RTS y CTS son frames pequeños que tienen información de quienes son las estaciones transmisoras, receptoras y cuanto tiempo durará la transmisión. En el caso de la estación oculta, A sensa el canal, si lo encuentra libre transmite un RTS a B indicando la longitud del frame que desea enviar. B responde con un CTS que también especifica la longitud del frame a recibir. En este momento C capta la respuesta de B, por lo que se percata de que va a tener lugar una transmisión en la que B actuará de receptor y sabe que deberá permanecer en silencio durante el tiempo que dure la transmisión (C sabe lo que durará pues conoce la longitud del frame y la velocidad de la red). Con esto, A envía los datos a B, y C puede transmitir a B una vez pasado el tiempo que él sabe debe esperar para poder comunicarse. En el caso de la estación expuesta B transmite a A un RTS indicando que quiere enviarle datos. En ese momento C se entera de las intenciones de B. A devuelve a B un CTS. Mientras tanto, C, ha captado el RTS pero no puede comunicarse con D, o al menos transferir los datos pues debe enviar primero un RTS, pero el canal estará ocupado pues debe esperar a que B termine. Si bien se soluciona correctamente el problema de la estación oculta, el sensar el medio hace que el problema no se resuelva (eficientemente al menos).

4.2.2. MACA

Multiple Access with Collision Avoidance o MACA es un protocolo MAC resuelve los problemas antes mencionados haciendo handshaking de señales RTS-CTS-DATA sin escuchar el canal, de ahí que el nombre sea MACA y que en ninguna parte de él lleve la parte CS. Cuando una estación tiene un frame que transmitir, antes de enviarlo, y sin escuchar el canal, envía un frame RTS, el nodo destino, al recibir el RTS, y si está en condiciones de recibir la transmisión, responde con un CTS. En el caso de la estación oculta ocurre lo siguiente: sin escuchar el canal, A transmite un RTS a B, y B responde con un CTS. C capta la respuesta de B, conociendo entonces que habrá una transmisión en la que B actuará de receptor, por lo que deberá permanecer en silencio durante el tiempo que dure la transmisión. A envía a B los datos correspondientes y una vez finalizado esto (pasado el tiempo) C puede transmitir a B. En el caso de la estación expuesta ocurre lo siguiente: B transmite a A un RTS indicando que quiere enviarle datos. En ese momento C se entera de las intenciones de B. A devuelve a B un CTS.

Mientras tanto, C, que ha captado el RTS pero no el correspondiente CTS, comprende que aunque detecta que B está transmitiendo el destinatario está fuera de su alcance, por lo que puede comunicar con D cuando quiera, sin esperar a que B termine. En este algoritmo también pueden ocurrir colisiones, como por ejemplo que choquen dos RTS vecinos. La solución se encuentra en que el o los nodos destinos no devolverán un CTS, por lo que pasado un cierto timeout, se implementará un algoritmo de retransmisión que permitirá a los emisores generar un nuevo RTS.

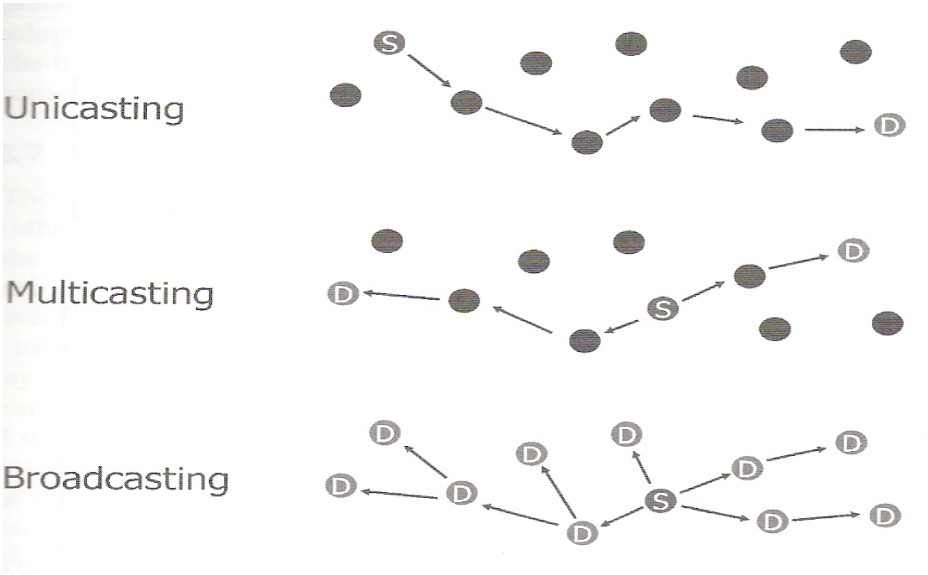
4.2.3. MACAW

MACA Wireless es una versión mejorada del protocolo anterior que funciona de manera similar, pero ahora utiliza un intercambio de mensajes RTS-CTS-DS (Data Send)-DATA-ACK(Acknowledge), además de implementar modificaciones al algoritmo de retransmisión o de backoff. La utilización de un frame de ACK en este nivel mejora los tiempos de respuesta, comparándolos con los que se obtendría si se dejara manejar la situación por el protocolo de nivel de transporte. El nuevo frame CS permite distribuir la información de sincronización sobre los períodos de contienda, de forma que los nodos puedan competir de igual forma por una fracción de tiempo para solicitar la transmisión. La transmisión se lleva a cabo de la siguiente manera, el emisor envía (sin escuchar el canal) un RTS al receptor, quien responderá con un CTS, una vez recibido éste, el emisor envía un DS seguido de los datos a transmitir. En caso de recibirse correctamente los datos el receptor devuelve un ACK, caso contrario no lo hace y se retransmite la información siguiendo el mismo esquema partiendo con el RTS. En el caso de que el ACK se pierda, se enviará un nuevo RTS al cual se le responderá nuevamente con el mismo ACK.

5. Enrutamiento en redes ad hoc

Por enrutamiento entendemos el conjunto de operaciones que realizan los distintos elementos que forman una red, desde el momento en el que un nodo origen desea enviar un paquete de datos a un nodo destino, hasta el momento en el que el mismo nodo origen ha recibido una confirmación de entrega de dicho paquete, o, por el contrario, hasta que el nodo origen desiste en el empeño de mandar dicho paquete, debido al fracaso de envíos iterativos y consecutivos del mismo.

Por ser una red Ad Hoc los elementos que forman dicha red van a ser los propios nodos de la red. El conjunto de las operaciones abarca operaciones de retransmisión de paquetes de control y de datos, cooperación con nodos vecinos, compartición de información de la red, así como distintas operaciones que según el protocolo pueden ser necesarias o no.



Recordemos, que en las redes multi-hop el nodo origen y el destino pueden estar separados por múltiples nodos, por ellos los datos deben ser reenviados a través de varios nodos hasta el destino. A continuación vamos a distinguir 3 tipos de enrutamiento en función del número de nodos a los que se de sea entregar un paquete. El primero, es el modo "unicasting", donde un nodo origen quiere enviar un paquete a un solo nodo, destino. Es en este tipo de enrutamiento en el que se ha basado nuestro estudio, pero vamos a comentar también los otros dos tipos, así como formas de enrutamiento para ambos tipos. "Multicasting" es el enrutamiento en el que un nodo desea mandar un paquete a un conjunto de nodos destino, este conjunto, lo denominaremos "conjunto multicast". Finalmente, está el caso en el que un nodo desea enviar un paquete al resto de nodos de la red, este tipo de enrutamiento es conocido como "broadcasting".

A la hora de clasificar los protocolos de enrutamiento se pueden tener en cuenta los diferentes aspectos:

- **Algoritmo de descubrimiento y mantenimiento de rutas.** En está clasificación se dividen los protocolos en función de la forma en la que se establece el

enrutamiento. En los protocolos proactivos cada nodo dispone información lo más actualizada posible sobre la topología y estados de los enlaces de la red, de tal forma que cuando se desea enviar o retransmitir un paquete, el nodo encargado de ello sabe en dicho momento a que nodo tiene que enviarle dicho paquete. Esto proporciona una rápida respuesta ante solicitudes de ruta y ofrece un buen comportamiento en situaciones donde la tasa de movilidad es alta. Por el contrario, para disponer de información coherente y actualizada de la red es necesario que un nodo esté mandando cada cierto tiempo paquetes de control a sus vecinos para almacenar toda información pertinente. Por tanto, la ganancia del tiempo de respuesta se ve contrarrestada con la disminución del ancho de banda del canal y la memoria y procesamiento de toda la información que se capta en cada momento.

En los protocolos reactivos o bajo demanda los nodos no disponen información sobre la topología de la red, si no que crean las rutas cuando es necesario. La sobrecarga de almacenamiento y procesamiento es mucho menor que en los anteriores, pero los retrasos de establecimiento de la ruta son mucho mayores. A favor de estos protocolos añadiremos la existencia de caches de rutas, es decir, pequeñas memorias donde un nodo va a almacenar distintas rutas ya establecidas (por el o por compañeros suyos de la red) Así, con el uso de estas caches se puede mejorar la latencia de respuesta,

Finalmente en este apartado añadiremos la reciente aparición de unos protocolos híbridos en los que se mantiene una filosofía proactiva en un ámbito local, y reactiva en un nivel más global. Ejemplos:

- Proactivos:
  - CGSR
  - DFR
  - DSDV
  - HSR
  - LCA
  - OLSR
  - TBRPF
  - WRP

- Reactivos:
  - Ad-hoc On-demand Distance Vector
  - Dynamic Source Routing
  - DYnamic Manet On-demand Routing
  - LBR
  - LMR
  - LUNAR
  - MOR
  - MPRDV
  - RDMAR
  - SSR
  - TORA

- Hibridos:

- HSLS
- ZRP

- **Reaparto de tareas entre los nodos.** En este caso la clasificación es mucho más sencilla. En los protocolos jerárquicos, los nodos pertenecen a diferentes niveles y su función en la retransmisión depende del nivel en el que esté. Normalmente, en el caso de este tipo de protocolos, las redes se dividen en grupos de nodos llamados *clusters*.

  Por el contrario, en los protocolos planos, todos los nodos están al mismo nivel, y, por tanto, tienen las mismas funciones y responsabilidades. Ejemplos:

  - CBRP (Cluster Based Routing Protocol)
  - CEDAR (Core Extraction Distributed Ad hoc Routing)
  - DART (Dynamic Address Routing)
  - DDR (Distributed Dynamic Routing Algorithm)
  - FSR (Fisheye State Routing protocol)
  - GSR (Global State Routing protocol)
  - HARP (Hybrid Ad Hoc Routing Protocol)
  - HSR (Host Specific Routing protocol)
  - HSR (Hierarchical State Routing)
  - LANMAR (Landmark Routing Protocol for Large Scale Networks)
  - OORP (OrderOne Routing Protocol)

- **Empleo de información geográfica.** Como su propio nombre indica, en los protocolos geográficos se tienen en cuenta la posición geográfica exacta de cada nodo para realizar los encaminamientos. Su gran inconveniente es la necesidad de que cada nodo ha de incorporar un dispositivo de posicionamiento global (GPS). Principalmente, esto supone dos cosas: una, un aumento significativo del coste de cada nodo de la red, y dos, un aumento, también bastante significativo, del consumo de la energía.

  Por el contrario, están los No geográficos que son aquellos protocolos que obvian la posición geográfica de cada nodo para establecer una ruta de comunicación entre los nodos. Ejemplos de protocolos geográficos

  - ALARM (Adaptive Location Aided Routing - Mines)
  - BGR (Blind Geographic Routing)
  - DREAM (Distance Routing Effect Algorithm for Mobility)
  - GLS(Grid) (Geographic Location Service)
  - LAR (Location-Aided Routing protocol)
  - GPSAL (GPS Ant-Like Routing Algorithm)
  - ZHLS (Zone-Based Hierarchical Link State Routing)
  - GPSR (Greedy Perimeter Stateless Routing)

- **Modo de almacenamiento de la información de enrutamiento.** Tanto en redes convencionales como en redes Ad-Hoc, los protocolos pueden clasificarse

atendiendo a dónde se guarda la información de encaminamiento. En el encaminamiento salto a salto (hop by hop routing) cada nodo decide solo la siguiente estación a la que enviará el paquete, atendiendo a sus propias tablas. Esto distribuye la complejidad por toda la red. En el encaminamiento en origen (source routing) la responsabilidad del encaminamiento recae sobre la estación origen del envío, que almacena en cada paquete la ruta completa que ha de seguir.

El encaminamiento en origen presenta una serie de ventajas claras:

- Se garantiza la ausencia de bucles, evitando que nodos intermedios con información desfasada hagan que los paquetes sigan un camino errático.

- Al nodo origen le resulta sencillo fijar rutas diferentes hacia un mismo destino sin necesidad de coordinarse con los nodos intermedios. Esto es útil para equilibrar la carga o para proporcionar diferentes calidades de servicio.

- Cada paquete incluye la ruta completa que ha de seguir, esto permite que se propague valiosa información de encaminamiento por toda la red, sin coste adicional.

A estas ventajas se contrapone un único inconveniente principal: la información de encaminamiento ocupa mucho espacio en las cabeceras de los datagramas, reduciendo el espacio disponible para datos.

Los protocolos para redes convencionales que emplean salto a salto pueden dividirse a su vez en dos grupos: Estado del enlace, como OSPF (Open Shortest Path First). Cada nodo conoce el estado de toda la red y luego calcula la ruta óptima al destino aplicando el algoritmo de Dijkstra. En otros protocolos de estado del enlace no se distribuye la información completa sino en forma estadística, centrándose en las zonas que parecen más interesantes. Vector de Distancias (DV, Distance Vector), también conocidos como (DBF, Distributed Bellman Ford), donde cada nodo lo único que conoce de la ruta para llegar a otro es el primer salto y la distancia hasta el destino. Estos algoritmos presentan el inconveniente del salto al infinito: supongamos una ruta que pase por los nodos ABCD. Supongamos que B quiere encaminar un paquete hasta D, pero el enlace BC está roto (tiene peso infinito). Entonces B intentará encaminar hasta D pasando por A, porque A le ofrece un camino. Pero no puede saber que ese camino precisamente acabará pasando por el mismo enlace roto, circunstancia de la que A no tiene noticia aún. Algunas técnicas paliativas para este problema son las conocidas como split-horizon y poisoned-reverse. Todo lo que se conoce sobre protocolos de red convencionales se puede aplicar para redes Ad-Hoc, lo que supone una gran ventaja. En situaciones poco exigentes, y como primera aproximación, podríamos aplicar cualquiera de los protocolos convencionales de Internet en una red Ad-Hoc, ya que son capaces de funcionar sin configuración inicial (son self-starting), se adaptan a cambios en la topología y ofrecen múltiples rutas para un destino. Pero es necesaria una adaptación, ya que precisamente el peor comportamiento de los protocolos convencionales se da cuando los nodos se mueven con frecuencia, por ejemplo exigen aplicar

continuamente el ya mencionado algoritmo de Dijkstra, que tiene complejidad exponencial.

En Internet, el encaminamiento se hace a partir de la direcciones. Históricamente se comenzó encaminando a partir de las clases, hoy se hace usando CIDR (Claseless Inter-Domain Routing). El principio es el mismo: a partir de cierto prefijo, de tamaño fijo o tamaño variable, se localiza el destino.

Para las redes Ad-Hoc inicialmente se intentó una aproximación similar, pero resultó inadecuado por ser muy costoso: en redes de este tipo no hay relación entre una dirección de red y una ubicación física, por tanto no sirve de nada agregar prefijos en las direcciones y la escalabilidad es un problema serio: las tablas de enrutado corren el riesgo de hacerse inmanejables, al exigir una entrada por dirección y no por grupo de direcciones. Otro problema es el del tráfico de gestión de la red: una red donde los nodos se mueven, aparecen y desaparecen, puede generar muchos mensajes de control. Es importante encontrar un equilibrio: demasiados mensajes consumirán el ancho de banda solo en mantener la red; un número muy bajo, hará la información de las tablas obsoleta.

Además, los mecanismos deben ser incrementales: sería muy ineficiente que una variación en un solo nodo obligase a recalcular la información de toda la red, puesto que los cambios son prácticamente continuos y las redes no tendrían apenas estados de estabilidad. Por todas las razones expuestas, se concluye que los protocolos de encaminamiento convencionales no son aplicables a las redes Ad-Hoc, que exigen algoritmos desarrollados específicamente para este entorno.

- **Número de rutas que se mantienen.** Esta es la clasificación más simple, y atiende simplemente al número de rutas que calcula y/o almacena un nodo. Un protocolo singlepath es aquel en el que solo se mantiene una ruta hacia cada destino. Por el contrario, los protocolos multipath son aquellos en los que se mantienen varias rutas hacia los distintos nodos destino.

5.1. Enrutamiento Unicast

5.1.1. Protocolos proactivos

Los protocolos proactivos mantienen rutas unicast entre todos los pares de nodos aunque no se estén usando todas las rutas. Por tanto, cuando lo necesita, una fuente de tráfico tiene una ruta disponible al destino deseado, lo que ahorra el retardo del proceso de descubrimiento de ruta. Estos protocolos también pueden encontrar rutas óptimas (cáminos más cortos) dado un modelo de coste de enlaces.

Los protocolos de enrutamiento en Internet (basados en vector de distancia como RIP o basados en estado de enlaces como OSPF), entran en esta categoría. Sin embargo, estos protocolos no son directamente aplicables a redes ad hoc por su escasez de recursos que impide asumir la alta sobrecarga además de sufrir algunos problemas de convergencia. Por tanto, se han propuesto algunas variaciones de estos protocolos para su empleo en MANETs. Estos protocolos se clasifican en dos categorías:

- Vector de distancia.
- Estado de enlace.

En los protocolos de vector de distancia, un nodo intercambia con sus vecinos un vector que contiene información sobre la distancia actual a todos los destinos conocidos. Esta información se propaga por toda la red y las rutas se calculan de manera distribuida. Por otro lado, en los protocolos basados en estado de enlace, cada propaga el estado de sus enlaces de salida a través de la red, formando actualizaciones de estado de enlace. Cada nodo calcula localmente las rutas de forma de descentralizada, empleando la información topológica completa.

5.1.1.1. Estado de enlace

Es esquema estado de enlace es similar a la versión centralizada del cálculo del camino más corto. Cada nodo mantiene una representación de la topología de la red con un coste para cada enlace. Para mantener estas representaciones correctas, cada nodo envía periódicamente mediante un broadcast el coste de enlace de todos los enlaces salientes a todos los nodos usando un protocolo como el flooding. Cuando un nodo recibe esta información, actualiza su representación de la topología de la red y aplica el algoritmo del camino más corto para encontrar el siguiente salto hacia el destino. Algunos costes de los enlaces en la representación pueden ser incorrectos debido al retardo de la propagación de la información, las particiones de la red, etc. Estas inconsistencias pueden dar lugar a la formación de ciclos. Sin embargo, estos ciclos tienen un período de vida corto, puesto que desaparecen cuando un mensaje de control llega a su destino.

5.1.1.1.1. Optimized Link State Routing (OLSR)

Fue creado por Thomas Clausen y Philippe Jacquet en el proyecto Hipercom INRIA. Se trata de un protocolo proactivo basado en la optimización de los clásicos protocolos link-state, basados en estado de enlace. OLSR funciona bien en redes con alto número de usuarios (nodos) y con una topología cambiante. Para llevar un control, se intercambian periódicamente mensajes de tal forma que se va aprendiendo la topología de la red y el estado de los nodos vecinos.

Como hemos visto, el intercambio de tantos paquetes, congestiona la red y supone un grave problema en las comunicaciones. Para solucionar esto, OLSR utiliza la técnica de MPR (Multi Point Relay). Gracias a esta técnica se reduce el número de retransmisiones. Veamos en qué consiste esta técnica. MPR (Multi Point Relay).

5.1.1.1.1.1.Multi Point Relay (MPR)

La técnica MPR consiste en seleccionar un mínimo conjunto de nodos vecinos a un salto de distancia, que sean capaces de llegar a todos los nodos vecinosque se encuentran a dos saltos de distancia.

De esta forma, un nodo selecciona su conjunto de nodos MPR, y sólo puede intercambiar mensajes de control con ellos. Así se evita el enviar de forma masiva mensajes de broadcast.

Para confeccionar la lista, cada nodo utiliza el mensaje "HELLO" que envía a todos los nodos vecinos. Este paquete tiene un campo conocido como tiempo de vida (TTL, Time To Live), que es de valor 1. Al tener el TTL un valor de 1, el mensaje sólo llega a los nodos que se encuentran a un salto de distancia y no es retransmitido por la red. De esta manera cada nodo puede conocer a sus nodos vecinos y a los vecinos de estos. De esta forma se puede saber que nodos conviene seleccionar como conjunto MPR.

En la Figura podemos ver como se selecciona un conjunto de nodos, MPR:

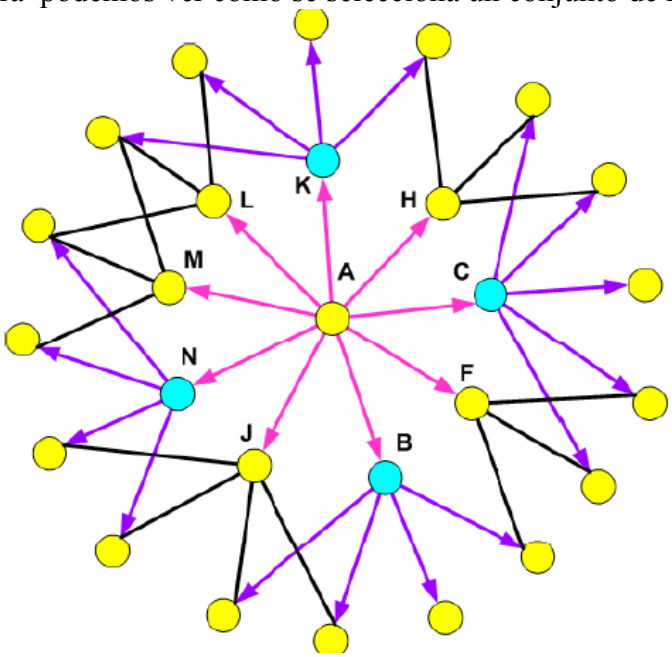


Figura 5.1. *Multi Point Relays*

Vemos en la Figura 5.1 como el nodo central (nodo A) selecciona el mínimo número de nodos a un salto de distancia (nodos B, C, K, N), capaces de llegar a todos los nodos que se encuentran a dos saltos de distancia. En el ejemplo el nodo A selecciona su lista de nodos MPR con los cuales sólo se enviará información evitando así la inundación de mensajes por toda la red.

5.1.1.1.1.2.Formato de paquete

Cuando un nodo recibe un paquete de encaminamiento OLSR determinará el procesamiento que debe seguir basándose en sus campos. A continuación se describen los más importantes:

- **Message Type:** En este campo se indica el tipo de mensaje que debe va a ser encontrado en la parte reservada para "MESSAGE" (campo de datos del paquete). En el caso de encontrarse vacío lo descarta.
- **VTime:** Este campo indica durante cuanto tiempo debe considerar la información que lleva el mensaje como válida.

- Message Size: Indica el tamaño del mensaje en bytes. Este campo se define al principio.
- **Time To Live:** Este campo contiene el número máximo de saltos que puede realizar un paquete. Sirve para no tener paquetes perdidos por la red y al mismo tiempo descongestionarla. Cuando se recibe un paquete con valor de TTL igual a 0, se elimina de la red. Si por el contrario el TTL es mayor a 0 se decrementa su valor en una unidad y se retransmite nuevamente.
- **Hop count:** En este campo se anota el número de saltos que ha realizado el paquete. Todos los nodos cuando lo recibe incrementan su valor en una unidad. Esto permite optimizar los recursos de la red evitando utilizar los caminos más largos entre origen y destino.
- **Sequence Number:** Cuando se crea un mensaje, se le asigna un número de identificación. Gracias a este número se puede saber si el mensaje de ha transmitido con anterioridad y así poder evitar retransmisiones.

5.1.1.1.1.3.Funcionamiento

Cuando un nodo recibe un paquete básico OLSR, analiza sus campos. Lo primero que hace es determinar de qué tipo se trata. Mira el campo Message Type, para determinar qué se encontrará en el mensaje.

A continuación mira el Message Size, para ver si el paquete es correcto o si por el contrario debe descartarlo. Se podría encontrar en el caso de recibir un mensaje vacío.

Mira el Message Sequence Number para saber si ha tratado ese mensaje con anterioridad o por el contrario se trata de uno nuevo. En el caso de no ser un mensaje repetido mira el valor del campo TTL.

Al analizar el TTL, decrementa en uno su valor; si el valor resultante igual a cero debe ser eliminado de la red.

Una vez analizado el paquete, mira la información del estado de enlace para poder encaminar hacia otros nodos con el fin de llegar a su destino.

Como se ha comentado antes, este protocolo va bien en redes con elevado número de nodos y con una topología muy cambiante. Esto ocurre porque se van intercambiando por toda la red mensajes de tipo TC (Topology Control). Con el mensaje TC cada nodo va actualizando sus enlaces con los vecinos y conociendo cualquier cambio en la topología de la red.

5.1.1.1.2. Topology Broadcast Based on Reverse-Path Forwarding (TBRPF)

Al igual que OLSR, TBRPF es un protocolo proactivo de estado enlace. Sin embargo, TBRPF emplea diferentes técnicas para reducir la sobrecarga. Los nodos TBRPF calculan el camino más corto a cualquier nodo de la red. Para reducir la utilización de ancho de banda, los nodos propagan sólo parte de su árbol a sus vecinos. TBRPF está formado por dos módulos principales:

- Descubridor de vecinos para mantener la información sobre el vecindario.
- Módulo topológico para descubrir la topología y calcular las rutas.

El modulo descubridor de vecinos permite a los nodos detectar a sus vecinos y determinar el tipo de conectividad de cada vecino. La conectividad puede ser bidireccional, unidireccional, y en el caso de una rotura de enlace, el enlace puede haberse perdido. Cada nodo envía periódicamente un mensaje Hello a sus vecinos. El mensaje Hello es differential, en el sentido de que sólo se incluyen cambios en el estado sus vecinos. Cada mensaje Hello contiene tres categorías de información sobre los vecinos. Un nodo puede ser clasificado como Neighbor request, neighbor reply o como neighbor lost. Estas categorías ayudan a los nodos a determinar la dirección de los enlaces con sus vecinos. En general, cuando un nodo *i* cambia el estado de su vecino j, incluye a *j* en la lista apropiada (indicando el nuevo estado del vecino) en tres mensajes Hello consecutivos. Esto permite que el nodo *j* pueda conocer su cambio de estado o declarará al nodo *i* como perdido si no escucha este número de mensajes Hello.

La lista de neighbor request contiene la dirección contiene la lista de los vecinos de los que se han recibido recientemente Hellos, pero para los que no se ha determiando aún que su enlace es bidireccional. Cuando un nodo i recibe un Hello de un nuevo vecino j, incluye a j en su tabla de vecinos y lo marca como unidireccional. La próxima vez que el nodo i transmita un mensaje Hello, i incluirá el nodo j en su lista de neighbor request de ese mensaje. Esto indica que i está solicitando que j confirme la recepción del mensaje Hello de i incluyendo a j en su lista de neighbor request. Si recibe este mensaje, j incluirá a i en su tabla de vecinos y lo marca como enlace bidireccional. En su siguiente Hello, el nodo j incluirá la dirección de i in la lista neighbor reply, indicando que el mensaje de i ha sido recibido y que existe un enlace bidireccional. El nodo i puede ahora actualizar su entrada para el nodo j y marcarlo como bidireccional. Para evitar la inclusión de enlaces transitorios, los nodos pueden esperar a recibir un número mínimo de mensajes Hello del vecino antes de crear una entrada nueva en la tabla de vecinos para ese nodo.

Una vez se ha creado una entrada en la tabla de vecinos para un nodo, el nodo puede monitorizar el estado del enlace para confirmar que la conectividad sigue existiendo. Si un nodo no recibe un número de paquete Hello del vecino j que supere un valor umbral, actualiza el estado del enlace a perdido. La próxima vez, i envía un mensaje Hello, que incluye la dirección de j en la lista de neighbor lost. Si recibe j el mensaje Hello, considera que la conexión bidireccional con i se ha perdido y cambia el estado de i en la tabla de vecinos a unidireccinal. Por otro lado, si j no recibe más mensajes Hello de i, elimina i de su tabla de vecinos y lo incluirá en la lista de neighbor lost en sus futuros mensajes Hello.

Para realizar el enrutamiento cada nodo TBRPF calcula el árbol de caminos más cortos desde él mismo a cada uno de los nodos alcanzables en la red. El árbol es calculado empleando una versión modificada del algoritmo de Dijkstra. Después de calcular el árbol, los nodos sólo transmiten una parte del árbol, llamada reportable subtree (RT) a los nodos vecinos. Para transmitir el RT se emplean dos tipos de actualizaciones topológicas. Se transmiten actualizaciones periódicas con el RT completo. Estas actualizaciones completas se utilizan para informar a los nuevos vecinos del contenido de RT y asegurar que toda la información topológica necesaria ha sido propagada. Con más frecuencia se envían mensajes más pequeños llamados

diffrential Los differential updates sólo transmiten los cambios en el RT que se han producido desde la última transmisión periódica. Para reducir el número de mensajes de contro, los mensajes de actualización de topología se combinan con los mensajes Hello para emitir un menor número de paquetes.

Para calcular RT, sea T(j) el subárbol del nodo i con raíz en el vecino j. Para cada vecino j, el nodo i incluye T(j) en su RT si y sólo si determina que uno de sus vecinos puede seleccionar a i como siguiente salto en su camino más corto a j. Para determinar esto, el nodo i calcula el camino con menos saltos de cada vecino a cualquier otro vecino, usando el identificador del nodo (ID) para evitar la creación de ciclos.

En la Figura , vemos un ejemplo de la formación de RTs en TBRPF.

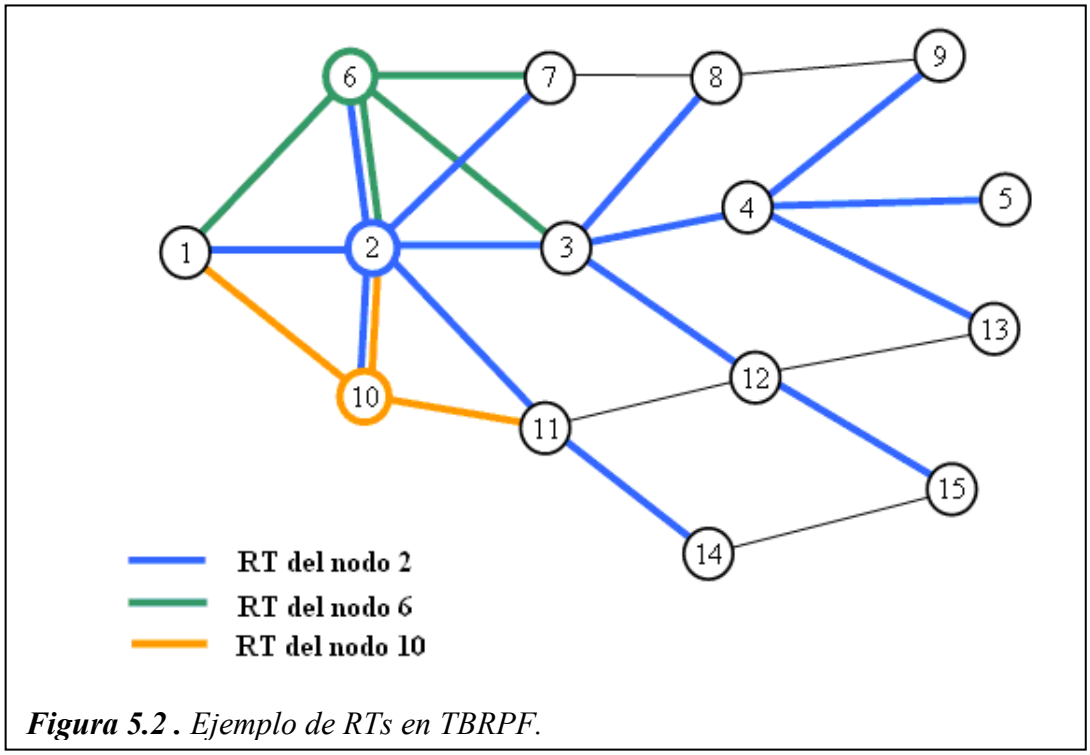


Figura 5.2 . *Ejemplo de RTs en TBRPF.*

5.1.1.1.3. Fisheye State Routing (FSR)

FSR es un protocolo jerárquico. Emplea la técnica "fisheye" propuesta por Kleinrock y Stevens, que es usada para reducir la cantidad de información necesaria para representar datos gráficos. El ojo de un pez captura con mucho detalle los píxeles cercanos al punto focal. El detalle decrece según aumenta la distancia al foco. En enrutamiento, el enfoque "fisheye" se traduce en mantener distancias exactas e información actualizada de los caminos con los vecinos cercanos, reduciendo progresivamente el detalle según aumenta la distancia.

FSR funciona de manera parecida a LSR en cuanto a que mantiene un mapa topológico en cada nodo. La diferencia principal es la forma en que se propaga la información de enrutamiento. En LSR, los paquetes de estado de enlace se generar y se propagan en cuanto un nodo detecta un cambio en la topología de la red. En FSR, los paquetes de estado de enlace no son propagados por toda la red. En vez de esto, los

nodos mantienen una tabla de estado de enlace basada en la información más reciente recibida de los vecinos, y que es intercambiada periódicamente con sus vecinos locales. Mediante este proceso de intercambio, las entradas de la tabla con número de secuencia altos, los sustituye por números de secuencia más pequeños. El intercambio periódico de tablas sustituir al intercambio periódico de vectores en DBF o DSDV. Sin embargo, aquí se propaga el estad del enlace y no un vector de distancia. Además, como en Link State Routing (LSR), se mantiene un mapa topológico completo en cada nodo, y se calcula el camino más corto usando dicho mapa.

En un entorno inalámbrico, un enlace de radio entre dos nodos puede sufrir frecuentes conexiones y desconexiones. El protocolo LSR realiza una actualización del estado de enlace cada vez que se produce un cambio, lo que al inundar la red causa una sobrecarga excesiva. FSR soluciona este problema realizando envíos periódicos, en vez de motivados por eventos, del mapa topológico, reduciendo la sobrecarga de paquetes de control. Cuando la red llega a alcanzar unas dimensiones considerables, los mensajes de actualización pueden a consumir demasiado ancho de banda, que dependerá de la frecuencia de actualización. Para reducir el tamaño de los paquetes de control sin afectar de manera importante a la exactitud de la información de enrutamiento, FSR usa la técnica “fisheye” aplicada a una red inalámbrica móvil, como se muestra en la Figura .

Los círculos con diferente nivel de gris, definen el diferente fisheye respecto al nodo central (nodo 11). El ámbito está definido como el conjunto de nodos que se pueden alcanzar en un determinado número de saltos. En este caso, se muestran tres ámbitos, a 1, 2 o m´s de 2 saltos. Los nodos están coloreados con los colores negro, gris y blanco respectivamente. El número de niveles y el radio de cada ámbito dependerán del tamaño de la red. La reducción de la sobrecarga de la red se obtiene de la diferente frecuencia de actualización para las diferentes entradas en la tabla.

Más concretamente, las entradas correspondientes a nodos dentro de un ámbito más pequeños son propagadas con mayor frecuencia. En la tabla de la Figura , las entradas en negrita se intercambian con más frecuencia. El resto se emiten con menor frecuencia. Como consecuencia de esto, el número de entradas en un mensaje de actualización se reducen. Esta estrategia produce que las actualizaciones de los nodos cercanos se produzcan puntualmente, pero puede ocasionar grandes latencias en los nodos lejanos.

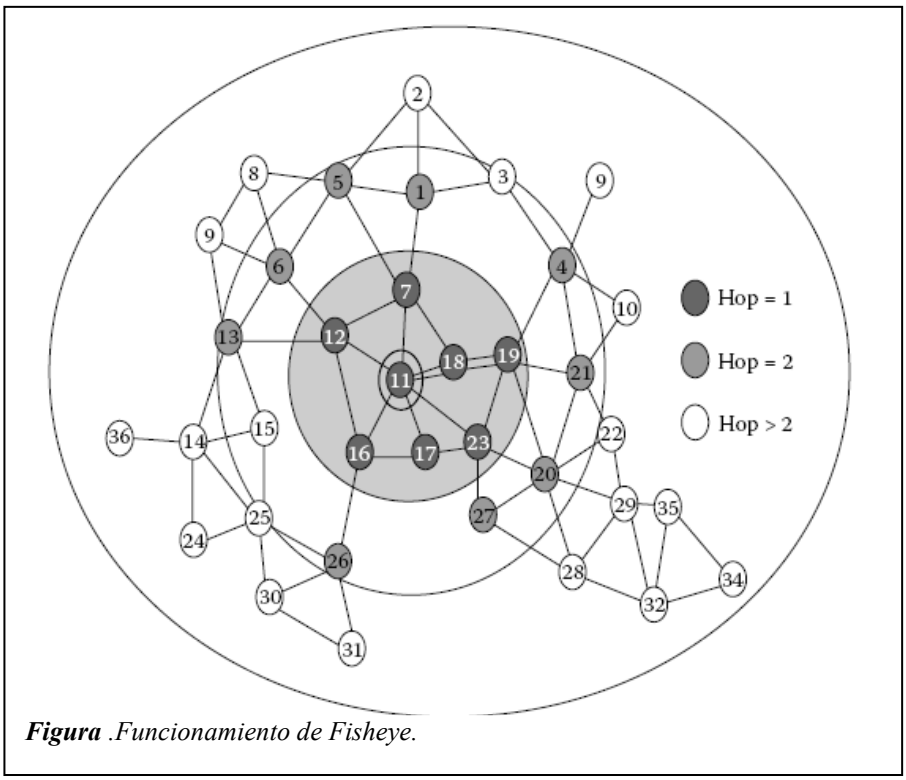


Figura *.Funcionamiento de Fisheye.*

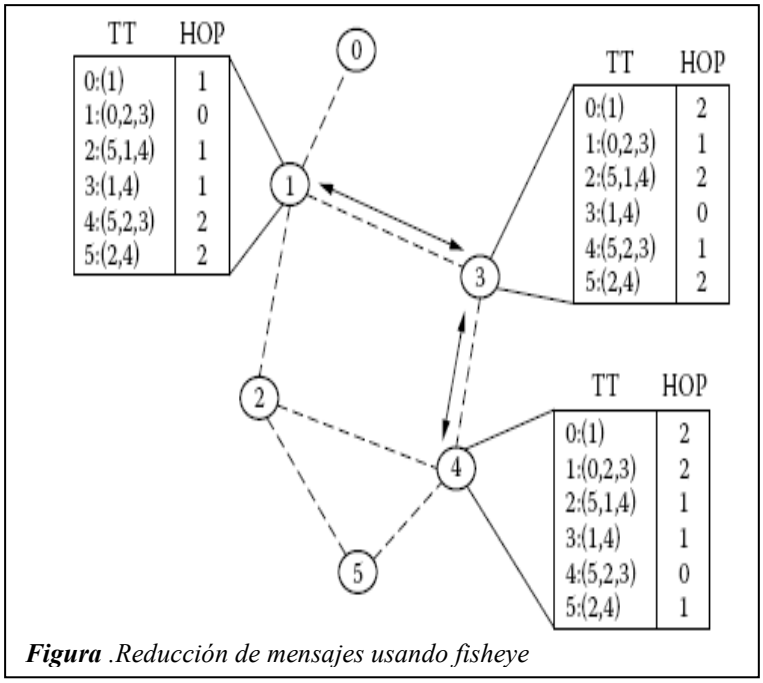


Figura *.Reducción de mensajes usando fisheye*

Sin embargo, el conocimiento impreciso del mejor camino a un destino lejano se compensa con el hecho de que las rutas irán mejorando poco a poco y los paquetes alcanzando su destino. Si la red llega a ser demasiado grande, se debe utilizar una estrategia para aumentar la frecuencia de dichos ámbitos para reducir la sobrecarga.

El concepto de FSR procede del protocolo Global State Routing (GSR). GSR puede ser visto como un caso especial de FSR, en el cual sólo existe un nivel de ámbito

fisheye. Como resultado, la tabla de topología completa se intercambia entre los vecinos. Esto consume una gran cantidad de recursos cuando la red se hace grande. Mediante la transmisión de la información de estado de enlace con diferente frecuencia según la distancia del ámbito, FSR logra una mayor escalabilidad manteniendo una baja sobrecarga con un compromiso entre la exactitud de la ruta y la cercanía del destino. Manteniendo una entrada para cada posible destino, FSR soluciona el trabajo estar de encontrar el destino (como en enrutamiento bajo demanda), lo que proporciona un retardo bajo en la transmisión de paquetes. Si la movilidad aumenta, la exactitud de las rutas a destinos lejanos puede disminuir. Sin embargo, según se acerca el paquete a su destino irá encontrando rutas mejores.

5.1.1.2. Vector distancia

En los algoritmos de vector distancia, cada nodo *i* mantiene, para cada destino *x*, un conjunto de distancias $\{d^x_{ij}\}$ donde *j* recorre todos los vecinos de *i*. El nodo *i* elige al vecino *k* como el siguiente salto para un paquete destinado a *x* si d^x_{ik} es igual $min_j\{d^x_{ij}\}$. La sucesión de siguientes saltos alcanzan *x* a través del camino más corto. Para mantener las estimaciones de las distancias al día, cada nodo monitoriza el coste de sus enlaces salientes y reenvía periódicamente a cada uno de sus vecinos su actual estimación de la distancia más pequeña a cualquier nodo de la red.

5.1.1.2.1. DSDV

Este protocolo fue uno de los primeros protocolos presentados para redes ad hoc. Su principal objetivo era conservar la simplicidad del RIP, manteniendo la ausencia de ciclos. Este protocolo se basa en el algoritmo clásico de vector distancia o Bellman – Ford, permitiendo calcular la ruta más corta a un destino.

Cada nodo en la MANET mantiene una tabla de enrutamiento con la siguiente información para cada destino:

- Dirección IP del destino.
- Número de secuencia del destino.
- Próximo salto (hop) al destino (dirección IP).
- Coste de la ruta hacia el destino (en número de saltos).
- Tiempo de instalación: Sirve para eliminar rutas antiguas.

Cada nodo envía periódicamente en modo broadcast su tabla actualizada a sus vecinos:

- Cada nodo añade su número de secuencia cuando envía su tabla de enrutamiento.
- Cuando los demás nodos reciben dicha información actualizan sus propias tablas de enrutamiento.

Las tablas de enrutamiento también pueden enviarse si se producen cambios en la topología de la red (creación o rotura de enlaces). En este caso, la información de actualización que viaja en los mensajes de enrutamiento es la siguiente:

- Dirección IP destino.

- Coste de la ruta hacia el destino (en número de saltos).
- Número de secuencia del destino.

Los nodos utilizan los números de secuencia del destino para poder distinguir entre antiguas rutas y rutas más recientes hacia un mismo destino. Un nodo incrementa su número de secuencia cuando se produce un cambio a nivel local en la topología de sus vecinos (se crea o se elimina un enlace). Aquella ruta hacia un destino que tenga asociada el número de secuencia del destino más reciente (el mayor) será la que se considerará válida. En el caso de que existan dos rutas con el mismo número de secuencia del destino, prevalecerá aquella cuyo número de saltos sea menor-

Se usan dos tipos de paquete de actualización de rutas:

- Full Dump: transportan toda la información contenida en la tabla de enrutamiento de un nodo. Este tipo de paquetes se envía muy raramente.
- Incremental: este tipo de paquetes transporta únicamente ela información contenida en la tabla de enrutamiento de un nodo que ha variado desde que el último paquete full Gump fue enviado. Estos paquetes son enviados con mayor frecuencia; así se evita una señalización y un consumo de ancho de banda excesivos debido al envío periódico de tablas de enrutamiento enteras y actualizadas.

Sin embargo, a pesar de la introducción de los paquetes "incremental", DSDV continúa teniendo problemas debido al exceso de señalización requerida, que crece de acuerdo con $O(N^2)$, siendo N el número de nodos de la red. Por esta razón, el protocolo no es escalable.

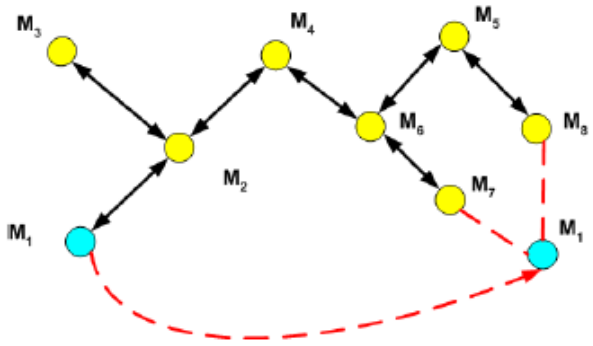


Figura 5.3. *DSDV.*

5.1.1.2.2. Wireless Routing Protocol (WRP)

El Wireless Routing Protocol creado ppo Shree Murthy y J. J. García-Luna Aceves es una de las primeras propuestas de protocolo de enrutamiento para MANETs. Fue propuesto en 1996 y el único protocolo que menciona es el DSDV.

WRP está basado en el algoritmo DBF. Los mensajes de actualización son enviados al conjunto de vecinos. Contienen toda la información de enrutamiento que el nodo conoce. Sin embargo, no se envía la tabla completa en cada actualización, sino que

se envía únicamente los cambios producidos o bien por la recepción de un mensaje de actualización de otro nodo, o bien por ruptura de un enlace.

WRP mantiene cuatro tablas:

- Tabla de distancia: indica el número de saltos entre un nodo y su destino.
- Tabla de encaminamiento: indica el siguiente salto hacia el destino.
- Tabla de coste de enlace: retardo asociado a una determinada ruta.
- Tabla de lista de retransmisión de mensajes: contiene un número de secuencia del mensaje de actualización, un contador de retransmisiones, un vector de confirmaciones y una lista de actualizaciones enviadas en un mensaje de actualización.

Los nodos envían mensajes de actualización periódicamente. El mensaje contiene una lista de actualizaciones, y una lista de respuestas indicando qué nodos deben confirmar la actualización. Un nodo envía un mensaje de actualización después de procesar los mensajes de actualización de sus vecinos o cuando detecta cambios en algún enlace.

Una parte original de este protocolo es la forma que tiene de encargarse de los bucles. En WRP los nodos anuncian la distancia y la información sobre el segundo salto para cada destino en la red inalámbrica. WRP pertenece a la clase de algoritmos de búsqueda de caminos pero con una importante excepción, evita el problema de cuenta hasta el infinito forzando a cada nodo a realizar comprobaciones de la información que le envía el nodo predecesor acerca de los vecinos.

5.1.2. Protocolos reactivos

Los protocolos de enrutamiento reactivos o bajo demanda son aquellos en los cuales los algoritmos de enrutamiento crearán rutas únicamente en el caso de que un nodo fuente necesite enviar información a un nodo destino. Así, se utilizan los recursos de red tales como la energía o el ancho de banda de forma más eficiente que en los protocolos de enrutamiento proactivos, aunque por otro lado, aumenta el retardo del Descubrimiento de Ruta.

Durante el proceso de Descubrimiento de Ruta, si un nodo fuente desconoce una ruta hacia el destino envía un mensaje de petición de ruta (Route Request) en modo broadcast para obtenerla y recibirá un mensaje de respuesta de ruta (Route Reply), que contendrá la ruta buscada.

Si los enlaces son bidireccionales y por tanto el mensaje que contiene la ruta buscada (Route Reply) puede utilizar la misma ruta que el mensaje de petición de ruta o Route Request, entonces la señalización introducida en el proceso de Descubrimiento de Ruta crece en el peor caso con $O(N + M)$ donde N representa el número de nodos de la red y M el número de nodos del camino de vuelta con la respuesta de ruta; para enlaces unidireccionales la señalización introducida crece con $O(2N)$.

Los protocolos de enrutamiento reactivos se pueden dividir en dos grupos:

- **Basados en la fuente (source-based).** Cada paquete de datos transporta en subcabecera la ruta completa de la fuente al destino, es decir, las direcciones de cada nodo intermedio a lo largo de la ruta desde la fuente al destino. Cada nodo intermedio consultará la cabecera del paquete que le llega para saber por dónde debe reenviarlo. Por lo tanto, ya no hace falta que cada nodo intermedio mantenga una tabla de encaminamiento con información actualizada continuamente mediante el envío periódico de mensajes de enrutamiento, como sucedía con los protocolos proactivos. Como contrapartida, en las redes ad hoc grandes, la probabilidad de que un enlace se rompa crece con el número de nodos y, además, al aumentar con mayor probabilidad el número de nodos intermedios a lo largo de la ruta, crece también la cabecera del paquete. En consecuencia, los protocolos de enrutamiento basados en la fuente no son recomendables en redes de gran tamaño con muchos saltos y alta movilidad debido a sus dificultades para escalar.

- **Salto a salto (hop-by-hop).** El paquete lleva en su cabecera únicamente la dirección del destino y la dirección del próximo salto, de forma que cada nodo intermedio a lo largo de la ruta en dirección al destino deberá consultar su tabla de enrutamiento para decidir por donde debe reenviar el paquete. La ventaja de utilizar este tipo de enrutamiento es que cada intermedio actualiza su tabla de encaminamiento continua e independientemente, de forma que cuando llega un paquete decide encaminarlo según el estado actual de la red y así las rutas pueden adaptarse más fácilmente a la topología dinámica de este tipo de redes. La desventaja de utilizar este protocolo es que cada nodo intermedio a lo largo de la ruta mantenga su tabla de enrutamiento permanentemente actualizada mediante el intercambio periódico de mensajes de actualización con sus nodos vecinos.

5.1.2.1. DSR

David B. Jhonson ideó en 1994 el DSR: Dynamic Source Protocol. Su objetivo era conseguir un protocolo sencillo y eficiente, que tratase de aprovechar al máximo las características de las MANETs.

Es un protocolo reactivo, bajo demanda, es decir no guarda información sobre el estado de la red, salvo rutas ya calculadas en "caches" destinados a ellas, es decir, la topología de la red varia sin que los nodos sepan la nueva configuración, pero, en el momento de una transmisión de un mensaje el nodo iniciará una serie de operaciones para averiguar la ruta hasta su destino. Como decimos, cada nodo dispondrá de una cache de rutas en las que puede almacenar distintas rutas, bien averiguadas o bien conocidas de alguna forma que comentamos a continuación.

DSR está compuesto por dos mecanismos, uno es el Descubrimiento de la Ruta (Route Discovery) y el otro es el Mantenimiento de la Ruta (Route Maintenance) Los explicaremos con más detalle, pero adelantamos que el primero se encarga de descubrir la ruta por la que viajarán los paquetes de un nodo origen a un nodo destino. El segundo verificará que la ruta escogida siegue siendo factible y que los paquetes están llegando al destino deseado de forma correcta.

5.1.2.1.1. Descubrimiento de ruta

Se produce cuando un nodo S quiere enviar paquetes de datos a un destino D, pero no conoce la ruta hacia D, entonces, S inicia este mecanismo. El nodo origen S empieza a mandar paquetes a sus vecinos, estos paquetes son denominados Route Request. Este paquete RREQ contiene el identificador del nodo origen y destino, e incluye también la ruta parcialmente calculada.

Cada nodo que recibe un RREQ va a añadir en la ruta contenida en el paquete su identificador y mandará a sus vecinos el nuevo RREQ. Este proceso se itera por los distintos nodos hasta que uno de estos paquetes lo reciba el destino. Este, al recibir el RREQ comenzará un sistema análogo mandando un Route Reply (RREP). Este proceso consiste en invertir la ruta calculada para hacer llegar al origen la ruta por la que viajarán los datos. En este proceso hay que tener en cuenta la direccionalidad de los nodos, ya que un nodo puede ser unidireccional o bidireccional. En el primer caso, un nodo A puede comunicarse y enviar datos a un nodo B pero este no puede comunicarse con A. En el segundo caso la comunicación se puede hacer en ambos sentidos. Hay que tener en cuenta esto porque al realizar el Route Reply se invierte la ruta calculada, por tanto, si uno de los nodos es unidireccional implica que no puede mandar el RREP al nodo correspondiente, y se ve obligado a iniciar otro Route Request para conseguir que el RREP llegue al nodo deseado y así proseguir con el Route Reply.

Recapitulando, uno nodo A quiere transmitir un mensaje a un nodo B. Primero se inunda la red con un Route Request, un mensaje que irá circulando por los distintos nodos de la red almacenando la ruta calculada. Cuando uno de estos paquetes llega al nodo B este responde al Route Request con un Route Reply, es decir, manda un mensaje que tiene que llegar al nodo A para informarle de la ruta por la que tienen que viajar los paquetes. El nodo A, al recibir el RREP introduce en su cache la ruta incluida en el propio RREP. Cuando A envía un mensaje de datos a D, toda la ruta que ha de seguir se incluye en la cabecera, de ahí el nombre de Source Routing. Los nodos intermedios usan la ruta incluida en la cabecera del mensaje para saber a que nodo mandar el mensaje.

En la siguiente figura vemos una posible configuración de los nodos de una red. De color azul vemos todos aquellos que reciben el Route Request, pero la ruta final por la que viajarán los datos será: S-E-F-J-D.

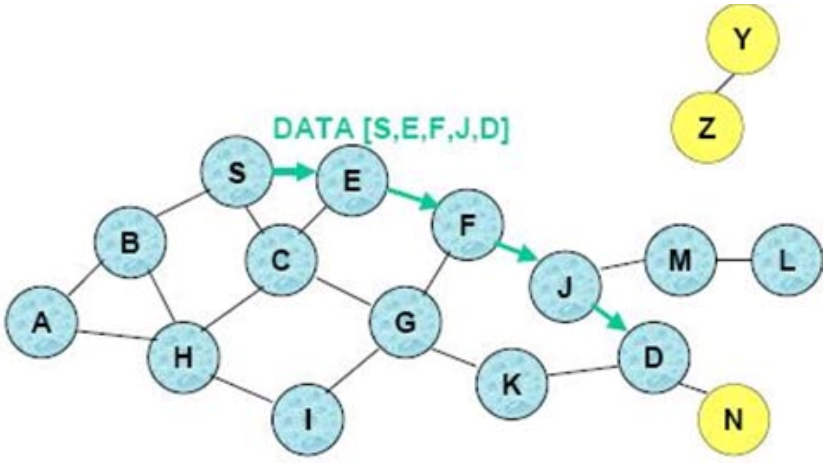


Figura 5.4. *Route Request.*

5.1.2.1.1.1.Caching Overhead

Hemos comentado que cada nodo va a disponer de una cache donde almacenarán las distintas rutas que conocen. Es importante este concepto ya que el conocer la ruta de antemano puede reducir el tiempo de entrega de los paquetes de sobremanera. Por tanto, tenemos que saber que rutas incluimos en la cache:

- **Paquetes "escuchados":** Un nodo A puede estar escuchando la retransmisión de un paquete de un nodo S a un nodo D sin que A intervenga en la retransmisión.
- **Ruta de envío en el paquete "fowarded":** Un nodo S está retransmitiendo un paquete a un nodo D usando otro nodo A entre medias, este nodo A en el momento de la retransmisión del paquete puede almacenar la ruta de esa retransmisión.
- **Ruta acumulada durante el Route Request:** Parecido al anterior solo que en este caso el nodo intermedio no retransmite un mensaje si no que ha recibido un Route Request con una ruta parcial y, aparte de proseguir el RREQ almacena la ruta que iba en la cabecera del propio RREQ.
- **Ruta acumulada en el Route Reply:** Análogo al anterior pero durante la propagación del RREP.

5.1.2.1.1.2.RREQ con Cache

Para ganar tiempo en las transmisiones ya hemos comentado lo importante que es el buen uso de la cache, en este apartado vamos a destacar otro aspecto muy importante del uso de la cache. Cuando se inicia un Route Discovery y un nodo recibe el RREQ mira si tiene en la cache el nodo destino, lógicamente, pueden darse dos casos, que el nodos se encuentre o, al contrario, que no se encuentre. Si no se encuentra se prosigue con el Route Request de forma normal, es decir, el nodo seguirá inundando la red con RREQs.

Por el contrario, si el nodo destino se encuentra en la cache del nodo *relay* (nodo intermedio) este nodo puede unir la ruta almacenada hasta el momento en el RREQ con la almacenada en la cache y cancelar el Route Request. De esta forma comenzaría directamente el Route Reply por parte del nodo *relay*. Es importante que se compruebe que no se repite ningún nodo en la nueva ruta generada. Por ejemplo, en la siguiente figura, el nodo A inicia un Route Request por que quiere mandar un paquete al nodo E. Suponemos que en la cache del nodo F tiene ya la ruta calculada desde F hasta E. En un determinado instante de tiempo F recibirá un Route Request con la ruta calculada hasta el momento, es decir, A-B-C-F. F comprueba su cache y como ya tiene la ruta calculada hasta el nodo E simplemente une ambas rutas, generando la siguiente ruta: A-B-C-F-C-D-E, como el nodo C está repetido en la secuencia (y produce un bucle bastante inútil) F antes de realizar el Route Reply, retoca la ruta y genera la A-B-C-D-E.

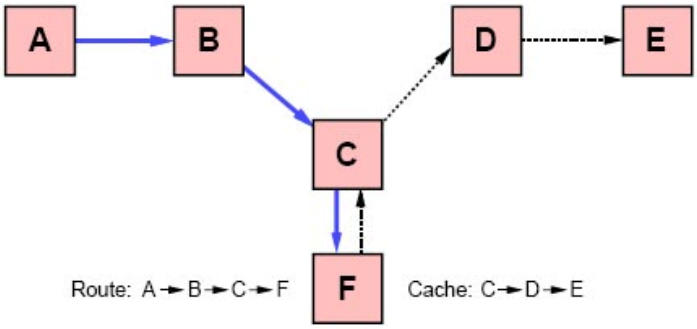


Figura 5.5. *Caché*

5.1.2.1.1.3.Route Reply Storms

No es extraño, ni mucho menos, considerar el caso donde varios nodos reciban un Route Request y varios de ellos tengan en su cache la ruta resultante. Comenzando por tanto, al mismo tiempo, varias Route Reply. Esto produce que varias transmisiones con el RREP sucedan al unísono malgastando el ancho de banda y aumentando las colisiones entre los paquetes.

Una posible solución, bastante sencilla, es que cada nodo espere un tiempo aleatorio en función del número de saltos hasta el destino. El nodo irá recogiendo poco a poco las rutas, de forma más ordenada y reduciendo las colisiones. Además, rechazará aquellas rutas más largas.

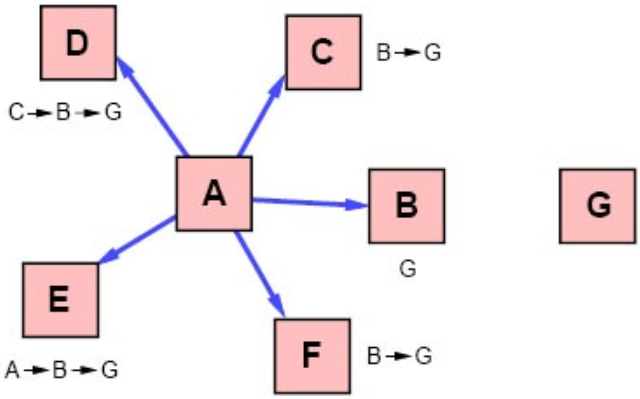


Figura 5.6.

5.1.2.1.1.4.Route Request: Límite de saltos

En el formato del paquete del Route Request se incluye un campo que indica el máximo número de saltos de un paquete RREQ. Este campo se decrementará en cada salto que se produzca durante el Route Request, y si llega a cero se deshecha el paquete.

Otra posibilidad es la emisión del RREQ de forma "*Anillo Expansible"*. Es decir el emisor inicia el Route Request con este campo igual a uno, si no recibe respuesta, dobla el campo y comienza de nuevo otro Route Request. Este proceso se itera sucesivamente hasta que, por fin, el nodo origen reciba el Route Reply.

5.1.2.1.2. Mantenimiento de ruta

Una vez finalizada la etapa anterior, es decir, el paquete de Route Reply ha llegado al origen con la ruta resultante, el origen comienza el envío de los datos por la ruta. Es en este momento cuando comienza la etapa de mantenimiento. Esta etapa comprueba constantemente que los paquetes llegan al destino deseado. Para explicarlo de forma más clara, vamos a observar la siguiente figura. En ese caso A está mandando paquetes a E. En cada tramo del envío un nodo se hace cargo de comprobar que el paquete que envía es recibido por el siguiente nodo. Es decir, A comprueba que B ha recibido el paquete, B comprueba que C lo recibe, etc.

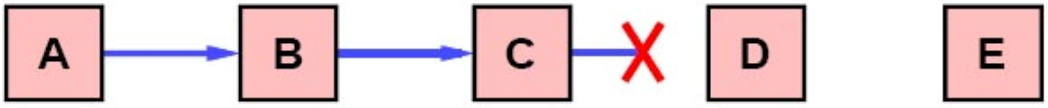


Figura 5.7. *Route Repair.*

Una posible forma de comprobar esto es "escuchando" las retransmisiones, es decir, para que A sepa si B ha recibido el paquete, A puede escuchar que B lo está retransmitiendo hacia C. Aún así se suele incluir un bit de comprobación en el paquete, para que el siguiente nodo responda un ACK para confirmar que ha recibido correctamente el paquete. En este caso hay que tener en cuenta la simetría de la conexión, es decir, controlar si el enlace es unidireccional o bidireccional. En caso de ser bidireccional no existiría ningún problema, pero, si fuera unidireccional el nodo que tiene que mandar la comprobación tiene que buscar rutas alternativas. Esto, como veremos, puede traer grandes inconvenientes.

Cualquier nodo implicado en la retransmisión de paquetes tiene que tener un sistema de control de envíos, se implementa de forma sencilla controlando el número máximo de envíos por paquete. En caso de que un nodo alcance esta cifra considerará que el siguiente nodo está fallando, y por tanto mandará un Route Error hacia el nodo origen. Los nodos actualizarán la cache para cambiar las rutas y que eviten ese enlace (ahora roto) El nodo origen tiene que reenviar la cache, tiene dos posibilidades, puede comprobar la cache para ver si tiene una ruta alternativa, al realizar el Route Request es muy probable que recibiera varios Route Replys alternativos. Si en la cache no tiene ninguna alternativa posible el nodo origen se verá obligado a iniciar otro Route Request y encontrar otra ruta.

5.1.2.1.2.1.Salvado de paquetes

Cuando un nodo falla en el envío hemos comentado una solución que consiste en avisar al nodo origen y que comience otro reenvío. Ahora bien, una posibilidad es que el nodo que falla en la entrega mande el Route Error hacia el origen, pero intente "salvar" el paquete, es decir, sería el propio nodo intermedio el que buscase una ruta alternativa. Así, cambiaría la ruta original del paquete por la nueva ruta escogida de la cache. Posteriormente marcaría el paquete como "*salvado*". Este marcado se realiza por si otro nodo posterior al que lo ha marcado fallase en la entrega y no existiera el marcado, podrían producirse bucles, y que, por tanto, el paquete nunca llegase a su destino.

Otra alternativa es dejar el prefijo de la ruta hasta el error y añadir un sufijo hasta el destino, así se evitan los bucles y el backtracking. Además, en este caso no se necesita el marcado del paquete.

5.1.2.1.2.2. Mejora de las rutas

Como hemos dicho al principio uno de nuestros principales objetivos es que sea un protocolo eficiente. Hasta el momento hemos comentado siempre la importancia de la optimización de la cache, además de esta forma existe un factor bastante obvio que reduce el tiempo de envío de los paquetes y es, obviamente, reducir el número de retransmisiones de un paquete.

Para ello, si durante el envío de los paquetes un nodo detecta una ruta más corta ese nodo actualizará la ruta para reducir el número de retransmisiones. Por ejemplo, en el caso de la figura el Route Request ha determinado que la ruta para entregar un paquete de A a D es A-B-C-D pero también se da el caso que A puede comunicarse directamente con C, por tanto, la ruta más corta es A-C-D. Este proceso automático de reducción de rutas se produce durante el Route Reply.

5.1.2.1.2.3. Propagación incremental de enlaces rotos

En caso de que se haya descubierto un error durante una transmisión de un paquete se propaga hacia el origen un Route Error. Una posible implementación es que el origen reenvíe el mismo route Error a sus vecinos en el siguiente Route Request que realice. De esta forma todas las rutas que se generen no contendrán en ningún caso el enlace roto.

5.1.2.1.2.4. Información corrupta en la cache

Para reducir el número de errores se puede almacenar en la cache información sobre enlaces rotos, nodos problemáticos. Se podría simplemente eliminar dicha entrada de la cache, pero de esta forma se garantiza que un Route Reply no contendrá en ningún caso información que ya ha generado errores (por tanto, con elevada probabilidad de volver a darlos).

5.1.2.1.3. Multicasting

Este modo de envío de paquetes no ha sido considerado en ninguna implementación del protocolo para ser resuelto de una manera eficiente. Las soluciones propuestas son soluciones sin control de información como pueden ser aquellas con uso de árboles o grafos multicast. Una manera poco efectiva de hacerlo es mediante un flooding con control de saltos y filtrado de paquetes. Consiste en limitar el número de saltos de un paquete por la red (véase el Hop Limits, explicado anteriormente) El filtrado, en cambio, consiste en que un nodo perteneciente a un conjunto multicast aceptará aquellos paquetes que vayan destinados a el. Pero, aquellos que no pertenezcan a ese conjunto los recibirán igualmente pero, al no pertenecer, harán caso omiso del paquete. Por eso no es eficiente, porque el paquete se manda a todos los nodos posibles sin distinguir durante los envíos si se envían a un destino adecuado o no.

5.1.2.1.4. Conclusiones

Cómo ya hemos repetido, es un protocolo reactivo, bajo demanda, es decir, no guarda información sobre la red (salvo alguna ruta en una “cache” destinada a ello) si

no que cuando desea enviar algún paquete procede a buscar la ruta hasta su destino. Esto es un arma de doble filo, como sabemos, la principal característica de una MANET es que sus nodos son móviles, así que el DSR al no guardar información de la red no requiere que se manden continuamente paquetes sobre el estado de la red. Ya que solo requerirá dicha información en el momento del envío (y ni si quiera información completa, si no tan solo la estrictamente necesaria para unir un origen con uno destino) A priori, esto suena muy interesante, ya que reduce la sobrecarga de la red y por tanto reduce el número de colisiones en la red (menos paquetes de control, menos mensajes circulando en un instante determinado, ergo, menos colisiones) Sin embargo, a posteriori la idea deja de ser tan atractiva, ya que, desde nuestro punto de vista, el número de inconvenientes que tiene conlleva la implementación sugerida es bastante mayor que las ventajas que trae. A saber, el tiempo de obtención de una ruta de datos puede ser vital, ya que en determinados casos (para comunicaciones en tiempo real principalmente) el "delay" de entrega de un paquete ha de ser mínimo.

Hemos comentado que la implementación admite tanto nodos direccionales como nodos bi-direccionales, esto de nuevo no es tan bueno como cabe esperar, ya que puede producir que un nodo al enviar un Route Reply no pueda comunicarse *directamente* con el nodo que tiene que recibir ese Route Reply, en cuyo caso debería de iniciar otra búsqueda de la ruta. Esto, indudablemente, aumenta de sobremanera el tiempo de entrega de un paquete.

Cabe destacar la importancia en este protocolo del papel tan importante que juega la cache de rutas de cada nodo. Hemos dicho que el tiempo de entrega de un paquete siempre debe de ser lo más pequeño posible, una cache correctamente configurada puede suponer grandes ganancias de tiempo. No almacenar nodos rotos pero sí almacenar el máximo posible número de rutas, tanto escuchadas, como efectuadas o simplemente direcciones que un nodo capta por hacer de "relay" en la retransmisión de distintos paquetes. Ahora bien, no podemos olvidar que en la MANET no solo son móviles los nodos, si no que, generalmente, también son de escasa memoria. Almacenar muchas direcciones implica un mayor tiempo de acceso a la cache (cuanto más grande sea más complicado y más retardo se obtiene en el acceso) y a la vez mayor es el espacio en memoria requerido.

En definitiva, es un protocolo sencillo y amigable, pero no del todo adecuado para redes grandes con alta movilidad, donde el rendimiento de la red se ve claramente perjudicado.

5.1.2.2. AODV

Fue creado por Charles E. Perkins como evolución de su anterior protocolo DSDV (Destination-Sequenced Distance-Vector). El DSDV inundaba la red de mensajes de control, de forma que la red se congestionaba y limitaba la duración de las baterías de los terminales. El AODV es uno de los protocolos más utilizados de los algoritmos reactivos, siendo idóneo para las redes Ad-Hoc. Este protocolo intercambia mensajes cuando necesita establecer una comunicación, es decir, envía mensajes a los vecinos para calcular cada ruta. Gracias a las mejoras incorporadas en AODV se evita la problemática de DSDV, pero por el contrario hay latencia cada vez que se calcula la ruta. Las características del protocolo:

- Señalización de control baja.
- Señalización de procesamiento mínima.
- Prevención de bucles.
- Funciona sólo con enlaces bidireccionales.

Cada nodo tiene asociada una tabla de encaminamiento que utiliza para poder establecer enlaces con otros nodos. Estas tablas de encaminamiento contienen los siguientes campos:

- Dirección IP Origen.
- Tiempo de Vida.
- Dirección IP Destino.
- Nº secuencia Destino.
- Contador de saltos (hop count).

Aparecen los campos de las direcciones IP de la fuente y de la IP del destino para saber en todo momento de donde vienen los paquetes y hacia donde han de ir.
También aparece un campo con el número de secuencia (del destino) que sirve para distinguir entre información nueva e información antigua y de esta forma evitar formación de bucles y transmisiones de rutas antiguas.

Otro parámetro que se almacena en las tablas de encaminamiento es el tiempo de vida. Este sirve para evitar que viajen paquetes perdidos por la red y utilizar enlaces de los que no se conoce su estado desde hace mucho tiempo. Cuando a un destino le llegan dos paquetes desde la misma fuente por caminos distintos, el campo hop count muestra el número de saltos que han tenido que hacer para cada una de las rutas. De esta forma se sabe cual de ellas es la ruta más corta y la que tiene que seleccionarse para hacer el envío de información.

Cada vez que se quiere comunicar una fuente con un destino, se inicia un proceso de descubrimiento de ruta, que finaliza cuando recibe un paquete con la ruta calculada. Existe otro concepto conocido como mantenimiento de ruta, que sirve para actuar en caso de que se rompa un enlace a lo largo de una ruta. Se consigue dando tiempo a las rutas descubiertas antes de considerarlas como invalidas.

5.1.2.2.1. Descubrimiento de rutas

Cuando un nodo quiere transmitir un paquete a un destino, lo primero que debe hacer es buscar en su tabla de encaminamiento a ver si existe una ruta hacia este destino previamente calculada. En el caso de encontrarla no iniciaría ningún proceso de descubrimiento de ruta, supondría que la que tiene almacenada en su tabla de encaminamiento es correcta y está actualizada. En el caso contrario, comenzará el proceso de descubrimiento de ruta (Route Discovery) para encontrar un camino válido.

El proceso comienza con el envío de un paquete RREQ (Route Request) en modo broadcast. Este paquete llega a los nodos vecinos que se encuentran a un salto de distancia y estos a su vez lo reenvían a sus vecinos y así sucesivamente hasta llegar al destino. Cualquier nodo que durante el proceso de búsqueda conozca la ruta hacia el destino, puede contestar con un paquete de RREP al nodo origen indicando la ruta que necesita.

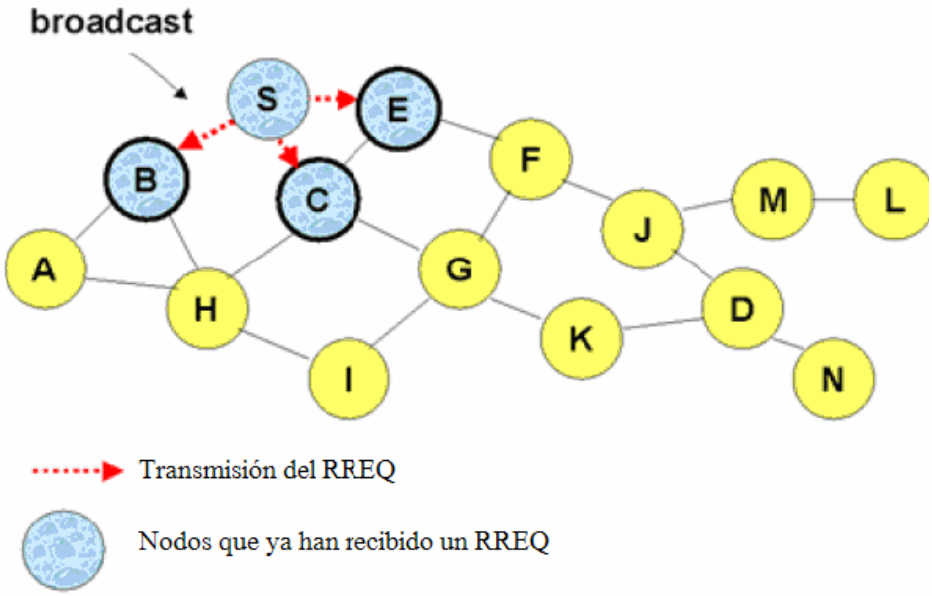


Figura5.8 *.Descubrimiento de Ruta. Iniciar el envío de datos del nodo S al nodo D.*

En el ejemplo de la figura 2.2, se quiere iniciar una comunicación entre el nodo S y el nodo D. Para ello el nodo S inicia un descubrimiento de ruta enviando un mensaje en modo broadcast a sus nodos vecinos. Estos nodos vecinos irán reenviando el mensaje hasta llegar al destino.

Todos los nodos mientras se va realizando el proceso de búsqueda, van actualizando las tablas de encaminamiento.

En el formato del paquete RREQ del protocolo de encaminamiento AODV, nos encontramos los siguientes campos:

- Dirección IP Origen.
- Número de Secuencia del Origen.
- Dirección IP del Destino.
- Número de Secuencia del Destino.
- RREQ Identificador.
- Contador de saltos (hop count).

Uno de los campos es el identificador que se va modificando cada vez que se genera un envío de RREQ. Esto sirve para que los nodos que lo vayan recibiendo (nodos intermedios) sepan si el paquete es idéntico al anterior (tiene el mismo identificador) y deben descartarlo, o por el contrario, si deben retransmitirlo (porque el identificador de paquetes es distinto).

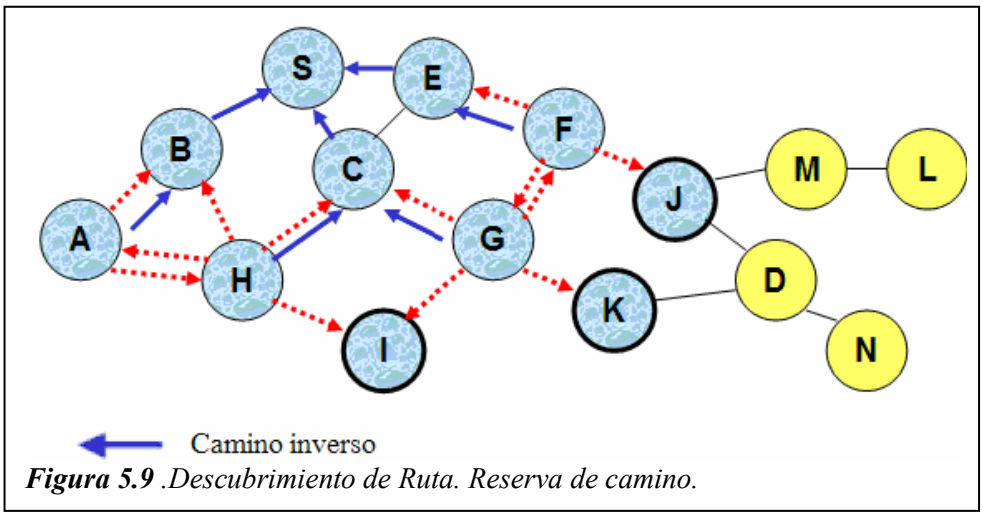


Figura 5.9 *.Descubrimiento de Ruta. Reserva de camino.*

En la figura 2.3 vemos como el nodo C vuelve a recibir el paquete en modo broadcast de los nodos H y G, pero detecta que el mensaje lo había recibido anteriormente y lo descarta sin reenviar nuevamente.

Cuando el mensaje llega al nodo destino, este responde al RREQ enviando de forma unicast un mensaje RREP (Route Reply). El mensaje RREP contiene la ruta hacia el origen invirtiendo el camino del RREQ.

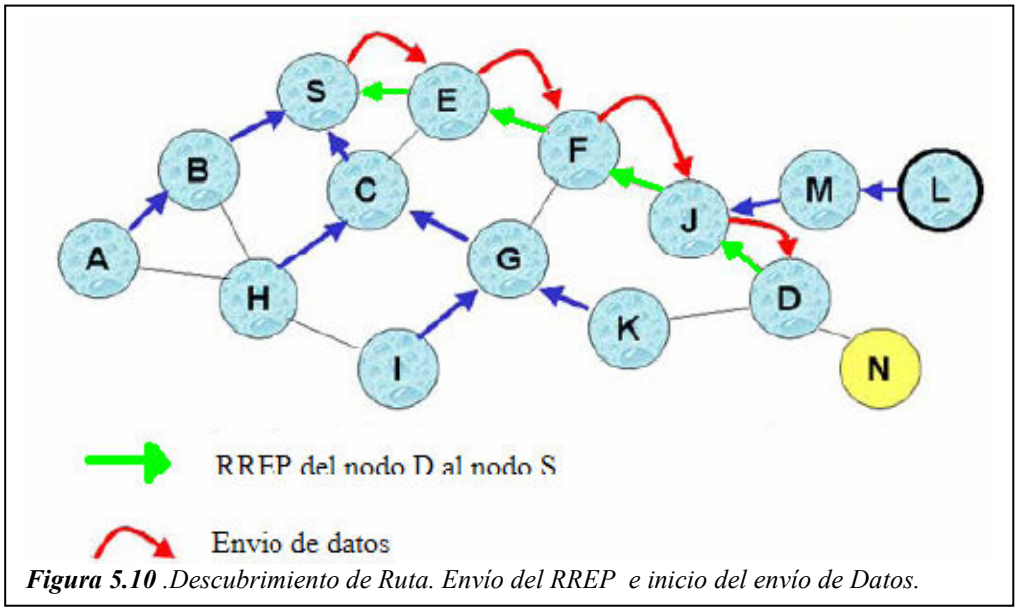


Figura 5.10 *.Descubrimiento de Ruta. Envío del RREP e inicio del envío de Datos.*

En la figura 2.4 vemos como el RREP sabe el camino hasta el nodo S al invertir la secuencia del RREQ de llegada. Una vez seleccionado el camino, ya se inicia el envío de datos.

5.1.2.2.2. Mantenimiento de rutas

Cuando una ruta es encontrada se le da un tiempo de vida y se considera útil hasta que este tiempo no expira. Esto se utiliza para no tener que iniciar un descubrimiento de ruta para cada mensaje de información que se quiere enviar.

Durante una comunicación entre el nodo fuente y el destino pude ocurrir que alguno de los nodos modifique su posición. Esto puede dar lugar a que se rompa el enlace y que la ruta quede inutilizada. El nodo vecino al enlace roto debe ser el encargado de informar al resto. Para ello se utiliza el envío del mensaje RERR (Route Error).

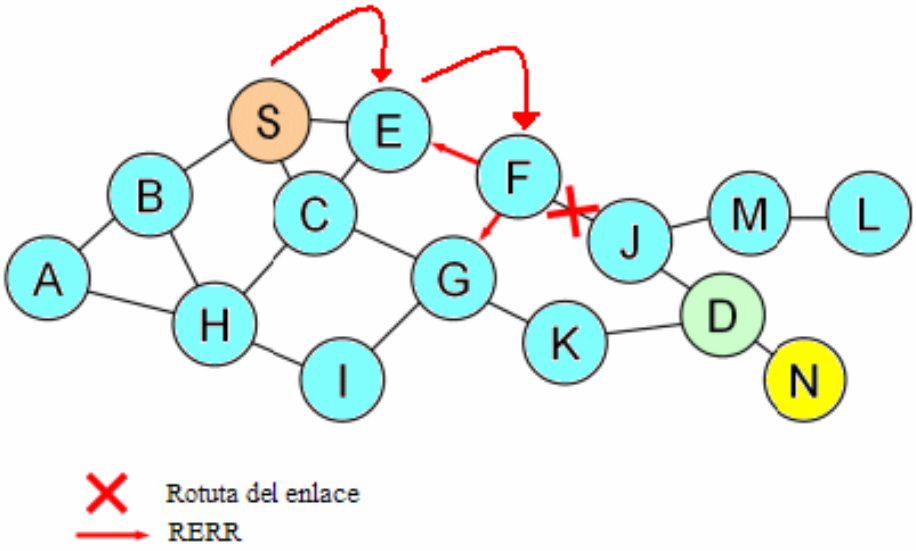


Figura 5.11. *Mantenimiento de Ruta. Envío del RERR.*

El mensaje viene a ser igual que el mensaje RREP pero con un número de salto (hop count) igual a infinito. Es decir, el nodo que detecta roto el enlace envía un RERR con valor de hop count hacia la fuente de valor infinito, lo que hace que cualquier otra ruta sea mejor y deban reencaminarse los paquetes por otro sitio. De esta manera, el nodo fuente decide si ha terminado la comunicación con el nodo destino o si por el contrario debe iniciar un nuevo proceso de descubrimiento de ruta.

Como vemos en la figura 2.5, el mensaje de RERR hace el camino invertido, de forma que recorre todos los nodos desde el F al nodo S. Así todos los nodos intermedios borran la ruta errónea, actualizando las tablas. AODV presenta una serie de opciones de optimización, como la posibilidad de reparar a nivel local un enlace roto que forma parte de una ruta activa.

Cuando se rompe un enlace, en lugar de enviar un paquete de RERR a la fuente, el nodo que ha detectado la rotura puede intentar repararlo localmente enviando un RREQ con el número de secuencia del destino incrementado en uno hacia ese destino. Los paquetes de datos se quedan almacenados en este nodo esperando recibir un RREP con una nueva ruta disponible hacia el destino. Si este nuevo procedimiento de Descubrimiento de Ruta no tiene éxito y el RREP no llega, entonces sí que será necesario informar a la fuente acerca de la rotura del enlace enviándole un paquete RERR.

5.1.2.3. DYMO

El protocolo DYMO (*Dynamic MANET On-demand*) es un protocolo reactivo que está en fase de desarrollo, por lo que actualmente no está estandarizado, pero está previsto que así lo sea en un futuro, tiene categoría de Standards Track. Hereda características de sus antecesores AODV y DSR, al ser un protocolo reactivo introduce una latencia al descubrir una ruta, pero lo contrarresta con el poco tráfico de control utilizado en la red. En este documento se explica el funcionamiento del protocolo y sus implementaciones. Se considera una evolución del AODV.

Es un protocolo joven que no está del todo desarrollado. La última versión del draft es la número 7, y está fechada en el 9 de febrero de 2007. Esta especificación está desarrollada gracias a Ian Chakeres de Boeing, Elizabeth M. Belding-Royer de la Universidad de Santa Barbara y Charles Perkins del grupo de investigación de Nokia.

5.1.2.3.1. Visión general

Es un protocolo joven que no está del todo desarrollado. En la actualidad se está creando la versión 4 del draft, ya que el último draft está fechado a día 23 de octubre de 2005. En esta última entrega aún no se especifica del todo como se realiza el mantenimiento de la ruta. Esta especificación está desarrollada gracias a Ian Chakeres de Boeing, Elizabeth M. Belding-Royer de la Universidad de Santa Barbara y Charles Perkins del grupo de investigación de Nokia.

Básicamente DYMO tiene dos mecanismos de forma similar a otros protocolos reactivos como DSR o AODV: el de descubrimiento de ruta, que se utiliza cuando en la tabla de encaminamiento no está la ruta del nodo destino, y el de mantenimiento de ruta, que puede utilizar varias formas para descubrir si una ruta se rompe. Hay tres posibles mensajes de control; el RE (*Route Element*) que engloba el RREQ (*Route Request*) para descubrir una ruta, y el RREP (*Route Reply*) para contestar el descubrimiento de una nueva ruta; el mensaje RERR (*Route Error*) que indica una ruta errónea y el UERR (*Unsupportedelement Error*) que es un mensaje de error no soportado por el protocolo. A éste último no le ha encontrado utilidad, por lo que en la última reunión del grupo de trabajo de MANET se ha decido suprimirlo del draft (*dicho cambio se verá reflejado en la versión del draft número 4*). El primero se envía en el descubrimiento de ruta y los dos últimos se utilizan para el mantenimiento de rutas. Dependiendo de la técnica de mantenimiento de las rutas empleada, también pueden existir mensajes de hellos o de reconocimiento, todos estos paquetes son enviados en UDP por un puerto To Be Determined4 (*TBD*). En el anexo B se muestra un resumen detallado de los formatos de los paquetes del protocolo.

5.1.2.3.2. Números de secuencia

El protocolo DYMO requiere que cada nodo de la red preserve su número de secuencia (*OwnSeqNum*) para asegurarse un mantenimiento de la red libre de bucles. Estos números de secuencia permiten que los nodos determinen el orden de descubrimientos de rutas, con lo que no permiten el uso de información caducada. Hay diferentes números de secuencia:

- El número de secuencia de la dirección del nodo destino de la tabla de encaminamiento de un nodo (*Route.SeqNum*).
- El número de secuencia del nodo destino cuando se envía un RE (*TargetSeqNum*). Si no tiene ningún número en su tabla, el valor que debe llevar es el de cero.
- El número de secuencia del nodo del bloque que transporta el RE
- (*RBNodeSeqNum*).
- El número de secuencia del nodo inalcanzable debido a que la ruta se ha roto (*UNodeSeqNum*), es transportado por el paquete RERR, en el caso que lo desconozca este será cero.

Para incrementar el OwnSeqNum se tiene que crear un RREQ o un RREP, y cumplir una de las dos condiciones siguientes:

- Que el TargetNumSeq sea superior al número de secuencia del nodo (*OwnSeqNum*).
- Que el TargetNumSeq sea igual al OwnSeqNum y el número de nodos por los que ha pasado un RE es menor al número de nodos intermedios por los cuales ha pasado un bloque hasta llegar al nodo.

5.1.2.3.3. Entradas en la tabla de enrutamiento

La tabla de encaminamiento es actualizada cuando se reciben paquetes de control del protocolo DYMO. Estos paquetes pueden contener información de una nueva ruta, actualización sobre un enlace o pueden indicar la ruptura de un enlace.

Cuando se recibe un RE y no se tiene constancia de la ruta especificada en el paquete, el nodo crea una nueva ruta. El nodo introduce en la tabla de enrutamiento los siguientes datos:

- La dirección IP del nodo destino.
- El número de saltos que hay entre el emisor y dicho nodo.
- El tiempo de vida de la ruta (*ROUTE_TIMEOUT*).
- La dirección del siguiente nodo hacia el destino.
- La interfaz por la cual reenvía los datos.
- El tamaño de la subred.
- El número de secuencia del nodo destino.
- Un indicador para saber si el nodo actúa como gateway.

En el caso de que una ruta exista en la tabla de encaminamiento se tiene que actualizar cuando:

- El temporizador no haya expirado y el número de saltos del paquete de control sea superior o igual al número que el nodo tiene en su tabla de encaminamiento.
- El temporizador ha expirado y el número del paquete es uno más que el que se tiene en la tabla.
- El número de secuencia del RE es superior que el que tiene el nodo.
- El número de secuencia del nodo destino en la tabla de enrutamiento no es conocido y se recibe un RE con un número de secuencia para ese destino.

Un nodo deja inactiva una ruta después de que pase un tiempo ROUTE_TIMEOUT, este temporizador se va reiniciando cada vez que el nodo reciba un paquete de datos y contenga esa ruta, ya que indica que está utilizando el enlace. Una vez que llegue a cero el temporizador ROUTE_TIMEOUT, la ruta pasa a estado inactivo durante un ROUTE_DELETE_PERIOD. Entonces, ésta no se puede utilizar y es borrada por el nodo cuando expira el temporizador. Cuando un nodo indica la ruptura de un enlace o le llega un paquete RERR indicando la ruptura de un enlace que utilizaba, el nodo tiene que poner el temporizador ROUTE_TIMEOUT de la tabla de encaminamiento de dicho nodo al final, así se obliga a que la ruta pase a estado inactivo.

5.1.2.3.4. Funcionamiento

El funcionamiento del protocolo DYMO se divide en dos mecanismos básicos; descubrimiento de ruta y mantenimiento de ruta. En el primero se explica cómo se puede descubrir una ruta en la red, y en el segundo se expone la forma que tiene el protocolo para detectar rupturas en las rutas. De esta manera los nodos descubren otra forma de llegar al destino.

5.1.2.3.4.1. Descubrimiento de ruta

Siempre que un nodo intenta enviar un paquete a un destino, el emisor comprueba que el destino esté en la tabla de encaminamiento. En el caso que ya exista esta ruta, el paquete envía la información al siguiente nodo basándose en la tabla. En el caso de que no esté, se realiza el mecanismo de descubrimiento de ruta.

Este mecanismo envía en modo broadcast el paquete RE indicando en un flag, que se trata de un mensaje de control RREQ. Cuando se crea este mensaje el OwnSeqNum se tiene que incrementar en una unidad. Los principales datos de este mensaje de control son:

- El TTL indica el número de saltos que le faltan para desechar el paquete, cuando es creado el valor es NET_DIAMETER.
- La dirección del nodo a la que se quiere enviar datos.
- El número de secuencia del nodo destino, en el caso que no se conozca este valor está a cero.
- Número de saltos por el que ha pasado el paquete (*THopCnt*).
- Estructuras de datos que informan del encaminamiento de una dirección (RBlock).

Cuando se crea un RREQ se tiene que crear el primer RBlock, el cual contiene:

- La dirección del nodo del RBlock al que pertenece.
- El número de secuencia del nodo que tiene la dirección en el bloque.
- El número de saltos que ha pasado este bloque.
- Un bit indicando si actúa como gateway.

El nodo que crea el RREQ se tiene que esperar RREQ_WAIT_TIME para poder enviar otro mensaje de RREQ. Para reducir una posible congestión en la red, éste tiene que seguir un tiempo de backoff binario exponencial, es decir, el primer intento de

RREQ tiene que esperar el tiempo RREQ_WAIT_TIME por dos, el segundo por cuatro y así sucesivamente. Se pueden hacer hasta RREQ_TRIES intentos antes de notificar que el nodo no es accesible. En el descubrimiento de la ruta el nodo emisor guarda los paquetes en un buffer, del que se borra la información si se agotan los intentos.

En el caso que le llegue un RREQ a un nodo intermedio y éste no disponga de la dirección del nodo destino, o el número de secuencia del RREQ (RBNodeSeqNum) sea superior al de su tabla de encaminamiento (Route.SeqNum), este paquete deberá actualizar la tabla de encaminamiento del nodo para realizar la ruta inversa. El nodo tiene que actualizar su OwnSeqNum sumándole uno, introducir un nuevo RBlock con sus propios valores, disminuir el valor del TTL y sumarle un salto al campo THopCnt. Una vez introducidos los cambios en el paquete se reenvía en modo broadcast.

Si un nodo recibe un paquete RREQ que contenga la dirección destino en su tabla de encaminamiento, entonces compara si el Route.SeqNum es superior al TargetSeqNum del paquete. En caso que no lo sea, se reenvía, ya que el nodo tiene caduca su tabla y actualiza los campos. Cuando es superior el nodo tiene que actualizar su OwnSeqNum sumándose uno y crear un paquete RE indicando que es RREP de respuesta al descubrimiento. Se introducen los saltos que le falta para llegar a dicho nodo, el número de secuencia del nodo destino, la dirección del siguiente salto y el RBlock del nodo que lo emite. Este nodo no debe enviar ningún paquete de RREQ al destino, por lo que el nodo destino tiene que hacer también un descubrimiento de ruta si desea realizar un enlace bidireccional.

Por último tenemos el caso que el nodo destino sea el que procesa el RREQ, en este caso el nodo compara si el OwnNumSeq es superior al TargetSeqNum, en el caso que no sea así se descarta el paquete. Contrariamente, si es superior, el nodo guarda los datos del RREQ en la tabla de encaminamiento para un posible enlace bidireccional. El nodo debe crear un paquete RE indicando que es una repuesta a un descubrimiento (*RREP*). Aumenta en uno el OwnSeqNum e introduce los datos en el paquete, este paquete se transmite al último nodo que ha transmitido el RREQ.

Una vez se envía el mensaje RREP por la ruta inversa, los nodos tienen que transmitirlo de forma unicast6 por la ruta inversa que le ha llegado el RREQ, estos nodos tienen que sumarle uno al OwnSeqNum y actualizar su tabla de enrutamiento. Cuando es recibido por el creador del mensaje RREQ, éste está preparado para transmitir los datos, estos son enviados vía el buffer.

5.1.2.3.4.2. Mantenimiento de las rutas

Este apartado es similar al descrito en el protocolo DSR. Cada nodo es el encargado de mantener el enlace del siguiente nodo tal y como se puede ver en la Fig.2.2. Cuando un nodo detecta la pérdida de un enlace, éste crea un RERR y lo transmite a los nodos anteriores de la ruta. Para el descubrimiento de una ruptura en una ruta, el draft propone cuatro posibles alternativas:

- Reconocimientos en la capa de enlace (*similar al usado en DSR*).
- Mensajes de Hellos: Este mecanismo no está descrito por el draft en la última versión.
- Descubrimientos de vecinos.

- Timeout de ruta: Este mecanismo no transmite el RERR. Una vez pasado un tiempo ROUTE_TIMEOUT la ruta queda inhabilitada, este tiempo va actualizándose cada vez que pasa un paquete por el nodo que utiliza dicha ruta.

Cuando un nodo descubre una ruptura, éste tiene que crear un paquete RERR, el cual tiene que introducir en el campo UNodeAddress1 la dirección del nodo inalcanzable, en el caso que se sepa el número de secuencia del nodo se introduce en el campo UNodeSeqNum. En caso que no se conozca, este campo toma valor cero. En el campo TTL se introduce el NET_DIAMETER y se envía en modo broadcast, si hay otra ruta que es perjudicada por la ruptura del enlace, se introduce también en el paquete RERR, en el cual se añaden nuevos campos de dirección y de números de secuencia.

Cuando un nodo recibe un RERR tiene que invalidar la ruta si:

- La ruta que invalida tiene como siguiente salto la misma dirección IP del paquete que ha transmitido el RERR.
- La ruta que invalida tiene como siguiente salto la interfaz del paquete que ha transmitido el RERR.
- El número de secuencia del nodo que nos ha llegado es cero o el resultado de restar el Route.SeqNum del nodo destino y el número de secuencia de la ruta inalcanzable del paquete es menor o igual a cero.

Si no pasa por algún filtro de éstos, la ruta inválida se retira del paquete y si el RERR contiene alguna ruta más se reenvía en modo broadcast. En caso que el paquete ya no tenga ninguna ruta inválida, éste ya no se transmitirá más. Si el paquete pasa todos los filtros, el nodo tiene que reenviarlo en modo broadcast bajando el campo TTL.

5.1.2.4. TORA

El Temporally Routing Algorithm (TORA) pertenece a la familia “link reversal”. TORA está diseñado para reaccionar eficientemente a los cambios topológicos y para tratar las particiones. El nombre del protocolo se basa en la suposición de la posesión de relojes sincronizados (por jemplo vía GPS) por parte de los nodos que es necesaria para ordenar los eventos que se producen en la red.

TORA enruta utilizando un esquema completamente diferente a los expuestos anteriormente. La optimalidad de las rutas es una tarea secundaria en TORA, cuyo principal objetivo es mantener rutas estables que puedan ser reparadas localmente de forma rápida. El protocolo construye un Grafo Acíclico Directo (DAG) con raíz en el destino deseado. El grafo se obtiene asignando una dirección lógica a los enlaces en base a una altura o nivel de referencia asignado a los nodos. Si (i,j) es un enlace directo del DAG, a i se le llama nodo upstream(antecesor) y j es el nodo downstream (descendiente). El DAG tiene la siguiente propiedad: sólo existe un nodo en el fondo, es decir, que no tiene enlaces salientes (el destino), mientras que el resto de los nodos tienen al menos un enlace saliente, normalmente más. El nodo destino puede ser alcanzado desde un nodo siguiendo sus enlaces salientes. El problema de los ciclos se resuelve por la propiedad del DAG. Podemos ver un ejemplo en la Figura .

El funcionamiento del protocolo puede ser dividido en tres fases: descubrimiento de ruta, mantenimiento de ruta y eliminación de rutas.

La primera fase consiste en el intercambio de paquetes de control Quero-reply. Durante la transmisión de los mensajes reply, el cual es enviado también por flooding, los enlaces reciben una dirección lógica (upstream o downstream) basada en su altura lógica relativa, por lo que se genera al final del proceso un DAG a partir del destino. Este estado de enrutamiento puede verse como una red de tuberías por las que fluye el agua descendiendo a través de la red hasta alcanzar el nodo más profundo (el destino).

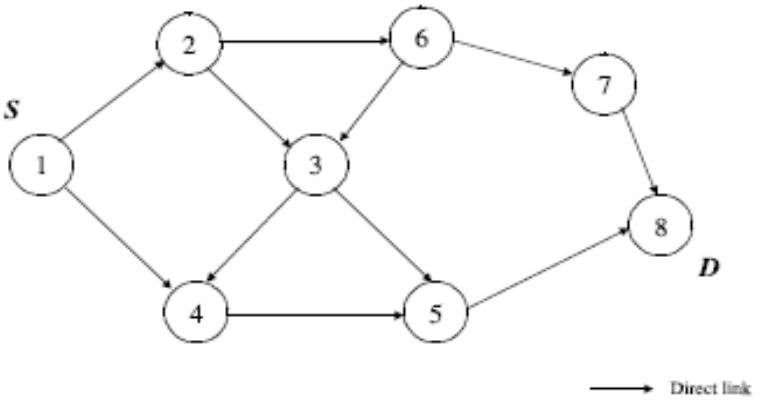


Figura 5.12. *Establecimiento de la dirección de los enlaces.*

El mantenimiento de ruta se activa para mantener el DAG actualizado ante los cambios topológicos de la red. Está basado en una secuencia finita de operaciones link reversal (giro del enlace). Una de las claves de TORA es que muchos cambios topológicos pueden no requerir ningún tipo de reacción. De hecho, si uno de los enlaces salientes de un nodo cae, pero ese nodo tiene al menos otro enlace saliente, el destino seguirá siendo alcanzable por otro camino, por lo que no se requerirá reparación de rutas.

Por otro lado, cuando un nodo detecta que no tiene nodos descendientes, genera un nuevo nivel de referencia que deberá ser el máximo global. El nuevo nivel de referencia será propagado por la red, provocando la ejecución de operaciones link reversal por parte de algunos nodos que, como resultado del nuevo nivel de referencia, han perdido todas las rutas al destino. Una vez finalizadas todas las actividades cercanas al nodo, el DAG ha sido reestablecido

TORA puede detectar particiones de la red. En este envía por flooding un paquete clear, que reinicia las tablas de enrutamiento para el destino.

Vamos a explicar más detenidamente el funcionamiento de cada mecanismo de este algoritmo y mostrar un ejemplo para aclarar algunos conceptos como el nivel de referencia que pueden resultar algo abstractos.

5.1.2.4.1. DAG

El DAG es una estructura de datos global que se forma a partir de las siguientes quíntuplas que mantienen cada nodo para cada destino conocido.

- **t:** momento en el que falla un enlace.
- **oid:** id del origen.
- **r:** bit de reflexión indica que es un nivel original si vale 0 y un nivel reflejado si vale 1.
- **d:** entero que ordena los nodos de forma relativa según el nivel de referencia.
- **i:** la id de los nodos.

La tripleta (t, oid, r) constituye el nivel de referencia. Y la tupla (d, i) es el offset (altura) en el sistema de referencia.

Como en el algoritmo Gafni-Bertsekas las alturas de los nodos para un destino dado determina la dirección de los enlaces del Grafo acíclico Dirigido. El DAG está orientado hacia el destino, estando las quíntuplas ordenadas en el sentido de que el destino siempre tendrá la menor altura, y el tráfico

Cada nodo mantiene una tabla de vecinos conteniendo la altura de los nodos vecinos. Inicialmente, la altura de todos los nodos es NULL, por lo tanto su quíntupla es (-, -, -, -, i). La latura del destino es (0, 0, 0, 0, dest).

5.1.2.4.2. Descubrimiento de rutas

Un nodo que necesita encontrar una ruta para un destino porque no tiene nodos downstream para dicho destino, envía un paquete QRY y activa el flan route-required. Un paquete QRY contiene la id del nodo para el que se está buscando una ruta. La contestación de un paquete QRY s denomina paquete UPD. Contiene la quíntupla del nodo vecino que contesta al QRY e indica en el campo destinatario el destino a quien corresponde dicha contestación.

Un nodo que recibe un paquete QRY realiza una de las siguientes acciones:

- Si el flag route required está activado, significa que no tiene que enviar el QRY, porque el ya ha lanzado un QRY para ese destino
- Si el nodo no tiene enlaces descendientes y el flan route-required no está activado, activa el flag y reenvía el mensaje QRY.
- Si un nodo tiene al menos un vecino descendente y la altura para este enlace es NULL, actualiza su altura al mínimo de las alturas de sus vecinos, incrementa el valor de d en uno y envía por broadcast un paquete UPD.
- Si el nodo tiene un enlace descendente y su altura no es NULL descarta el paquete QRY si ya se ha enviado un paquete UPD desde que se activó este enlace (activó el flan route-required). En caso conrario envía un paquete UPD.

Si se recibe un paquete UPD se actualiza en la tabla la altura del vecino y se realiza una de las siguientes acciones:

- Si el bit de reflexión del vecino no está activado y su flan route required está activado, toma como valor para su altura la de sus vecinos y la incrementa en uno. Luego elimina el flag route-required y envía un mensaje UPD a sus vecinos que ya pueden enlutar a través de él.

- Si la ruta de los vecinos no es válida (lo que viene indicado en el bit de reflexión) o el flan de route-required fue desactivado, el nodo sólo actualiza la entrada de sus vecinos en la tabla.

Vamos a ilustrarlo con un ejemplo. La red está formada por ocho nodos. Los nodos con un círculo son nodos con el flan route-required activado.

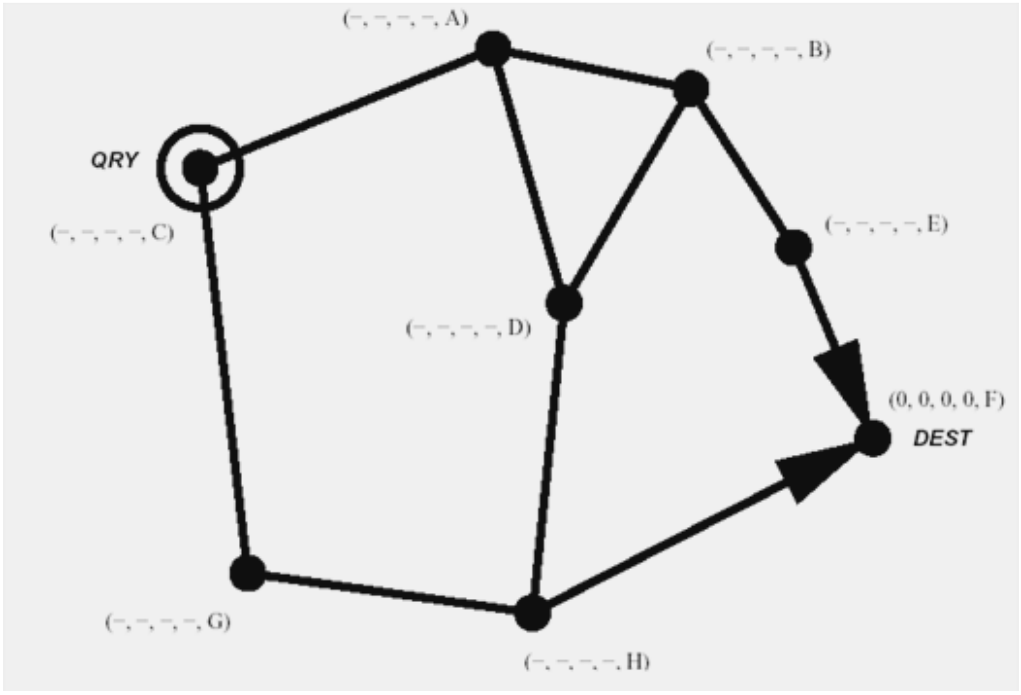


Figura 5.13. *El nodo C quiere comunicarse con F.*

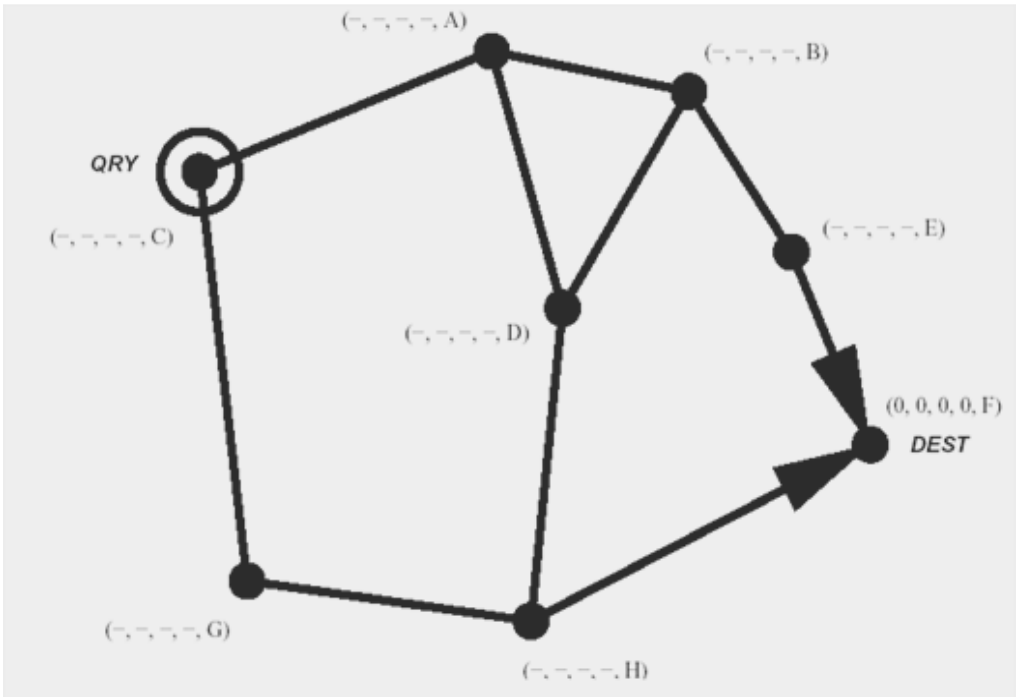


Figura 5.14 . *El nodo C envía un QRY.*

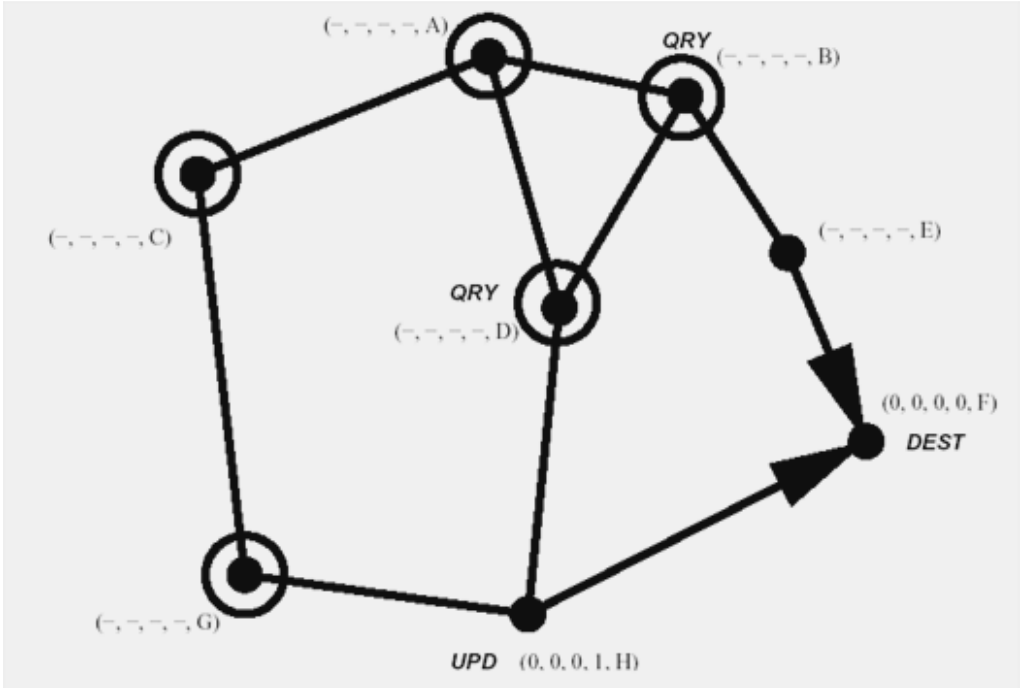


Figura 5.15. *H tiene un enlace con un vecino.*

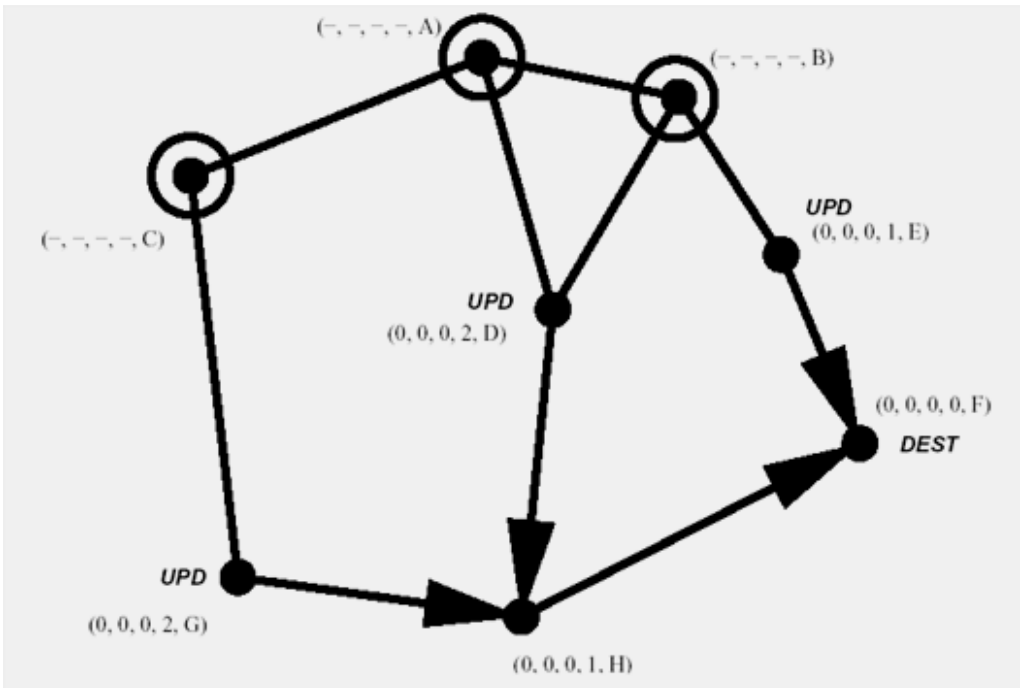


Figura 5.16.

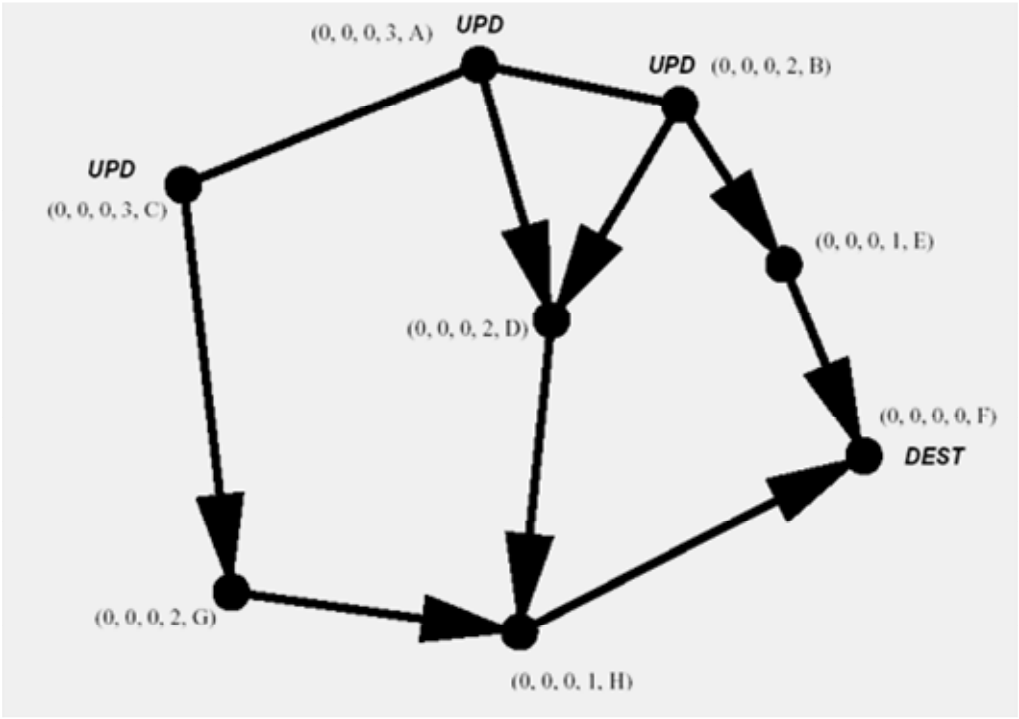

(0, 0, 0, 3, A)
UPD
UPD
(0, 0, 0, 2, B)
UPD
(0, 0, 0, 3, C)
(0, 0, 0, 1, E)
(0, 0, 0, 2, D)
(0, 0, 0, 0, F)
DEST
(0, 0, 0, 2, G)
(0, 0, 0, 1, H)

Figura 5.17.

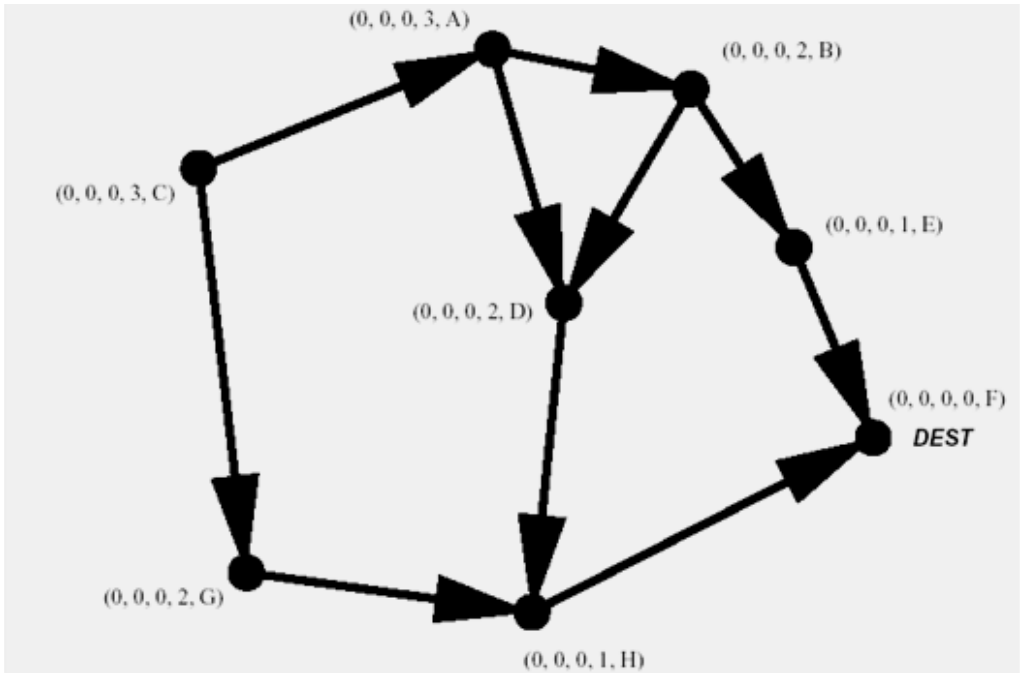


Figura 5.18.

5.1.2.4.3. Mantenimiento de rutas

El mantenimiento de ruta tiene cinco casos:

Generar: El nodo que ha perdido su último nodo downstream debido a la ruptura del enlace. El nodo define un nuevo "reference level", por lo tanto pone en el campo oid su id y en t el momento del fallo. Esto lo hacen los nodos que tienen vecinos upstream, si no ponen su altura a NULL.
Propagar: El nodo no tiene más enlaces downstream debido a la inversion de un camino siguiendo a la recepción de un paquete update y los niveles de referencia (t, oid,r) de sus vecinos no son iguales. El nodo propaga el nivel de referencia más alto entre sus vecinos y asigna al offset un valor (-1) que es más bajo que el offset de sus vecinos con nivel máximo.
Reflejar: El nodo ha perdido sus enlaces downstream debido a la inversion de un enlace seguido de la recepción de un paquete de actualización y la altura de referencia de sus vecinos son iguales con el bit de reflexión no activado. El nodo refleja después la altura de referencia.
Detectar: El nodo ha perdido sus nodos downstream debido a una inversion de un enlace seguido de la recepción de una paquete de update y la altura de referencia de los nodos vecinos son iguales con el bit de reflexion activado. Esto significa que el nodo ha detectado una partición y va a comenzar la eliminación de una ruta, Los valores de altura se configuran a NULL.
Generar: El nodo ha perdido su ultimo enlace downstream debido a la inversion de un enlace seguido de la recepción de un paquete update y la altura de sus vecinos es son iguales con el bit de reflexion activado y el oid de los nodos vecinos no es el id del nodo. El nodo luego configura t con el momento del fallo de enlace y oid con su propia

dirección. El valor de d es puesto a 0. Esto significa que el fallo de enlace no require reacción.

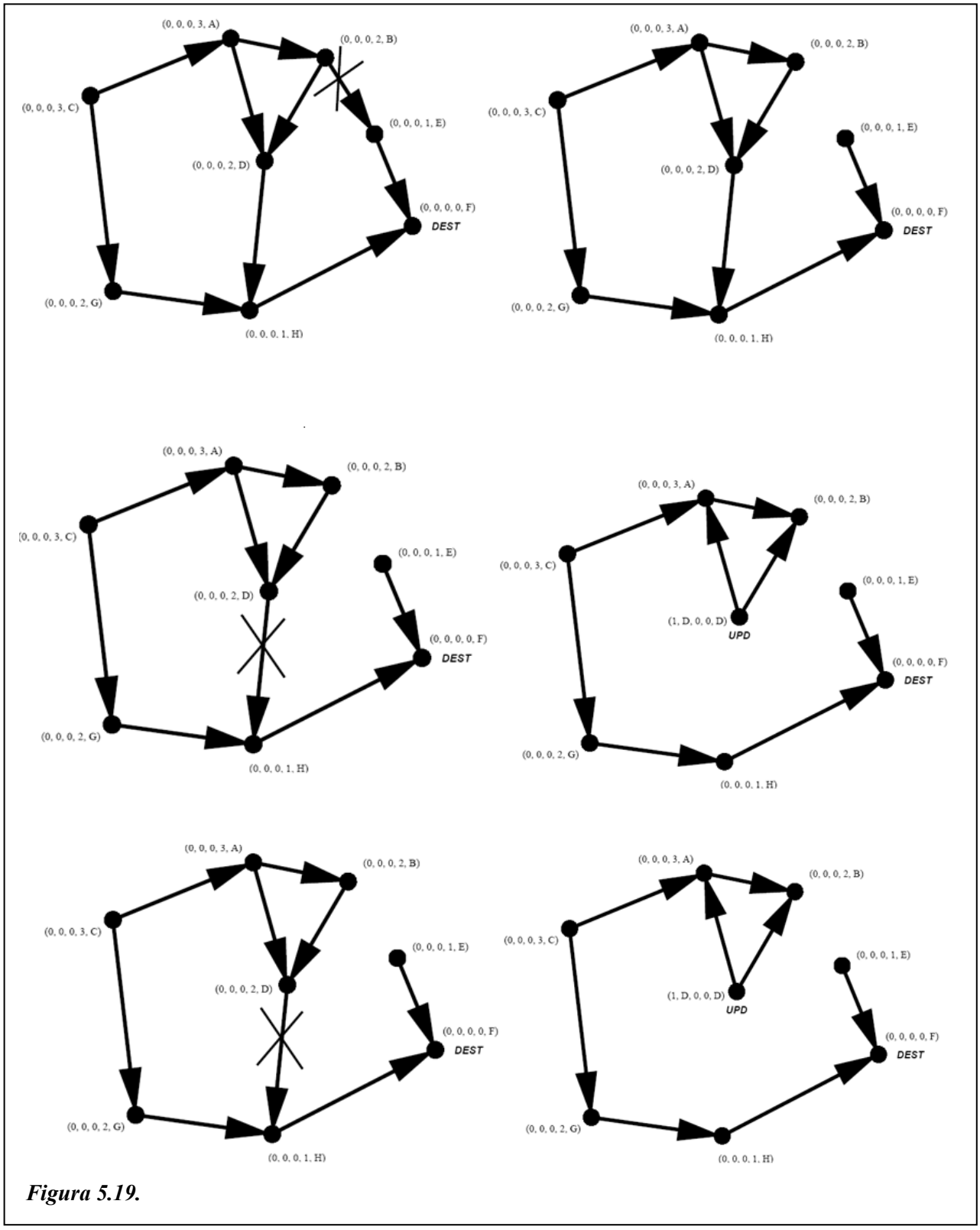


Figura 5.19.

5.1.2.4.4. Particiones y eliminación de rutas

Cuanto un nodo detecta una partición de la red pone su altura y la altura de sus vecinos para el destino a NULL y envía un paquete Clear. El paquete CLR consiste en nivel de referencia reflejado (t, oid, 1) y el id del destino. Si un nodo recibe un paquete CLR y el nivel de referencia coincide con su propio nivel de referencia pone sus alturas y las de sus vecinos para el destino a NULL y envía un paquete CLR en modo broadcast.

5.1.2.4.5. Conclusiones

La partición de la red es detectada en dos pasos del algoritmo para los nodos afectados. La eliminación de ruta necesita tres fases. Comparado con LMR que emplea una cantidad incierta para eliminar las rutas inválidas, mientras que TORA converge en exactamente tres fases de los nodos afectados. Si la red se hace grande y la movilidad es alta, la sobrecarga de los mensajes para las inversiones de enlaces propagándose por toda la red puede ser demasiado elevada TORA supone que existe un reloj sincronizado para todos los nodos.

5.1.2.5. Associativity Based Routing Protocol (ABR)

El Associativity Routing Protocol (ABR) es un protocolo reactivo diseñado especialmente para trabajar en entornos móviles. La idea clave es el uso de la longevidad de las rutas, en vez de la longitud, como el criterio principal de selección. En ABR, una ruta con una vida larga es preferida a una con menor vida, incluso aunque su longitud sea menor. Haciendo esto, el protocolo emplea siempre las rutas más estables, lo que en principio ahorra trabajos de mantenimiento. Un esquema similar es el propuesto por el Signal Scalability Protocol (SSA). La estimación de la vida de una ruta incluye la combinación de varias medidas sencillas (tiempo restante de batería, potencia de la señal, etc) y una nueva métrica, la asociatividad entre nodos, que mide el grado de estabilidad de un nodo respecto a otro a lo largo del tiempo y el espacio.

La técnica propuesta para medir la asociativita es la siguiente. Cada nodo genera una señal periódica y cuenta las señales recibidas de sus vecinos para actualizar sus "associativity ticks". Los associativity ticks son reiniciados si se deja de recibir la señal de un vecino durante cierto período de tiempo.

El valor de los associativity ticks permite clasificar un nodo en función de su alta o baja mobilidad respecto a sus vecinos. Un valor alto del nodo i respecto al nodo j indica que i fue capaz de recibir muchas señales consecutivas de j. Por tanto, el protocolo supone que es positivo que dos nodos permanezcan unidos. Por tanto, el nodo muestra un estado de baja movilidad con respecto a j, así que el enlace entre i y j es clasificado como long-lived. En cambio, un valor bajo para los associativity ticks indica una vecindad transitoria y, por lo tanto, un estado de alta movilidad.

El protocolo consiste en tres fases:

- Descubrimiento de ruta.
- Reconstrucción de ruta.

- Eliminación de ruta.

Como en otros protocolos reactivos, el descubrimiento de ruta está basado en el flooding. Los nodos intermedios no tienen permitido contestar una solicitud de ruta. El camino es seleccionado por el nodo destino y es almacenado como el camino activo en los nodos intermedios de modo hop-by-hop.

Un nodo fuente adquiere una nueva ruta a un nodo destino enviando en modo broadcast un paquete de control Boradcast Query (BQ). Al propagar el paquete, los nodos intermedios añaden su propio ID y su calidad como routers, que incluye los associativity ticks.

El nodo destino espera un determinado período de tiempo después de recibir el primer paquete BQ ya que puede recibir otros paquetes de solicitud enviados a lo largo de otros caminos. De esta forma, el nodo puede elegir el mejor camino de acuerdo en el siguiente criterio. Si una ruta está compuesta de nodos con un valor alto para los associativity ticks, se elige esa ruta en lugar de otra con menos saltos. El númeo de saltos sólo se tiene en cuenta para seleccionar entre dos rutas con un grado de asociatividad similar.

Una vez se ha seleccionado una ruta, el destino envía un paquete de control BQ reply al emisor a través de la ruta seleccionada. Como el BQ-reply es transmitido marcha atrás hacia la fuente, los nodos intermedios envueltos en su retransmisión pueden crear una entrada para la ruta, activando de esta manera la ruta de envío.

La fase de reconstrucción de ruta (RRC) se emplea cuando se viola la estabilidad de la asociación. En algunos casos el procedimiento puede intentar reparar los subcaminos corruptos, iniciando un descubrimiento del subcamino, sin notificar la ruptura a la fuente. En el peor de los casos, se envía un mensaje de control Route Notification (RN) a la fuente que inicia una nueva fase de descubrimiento de ruta. La última fase del ABR es la fase de Eliminación de Ruta. Esta fase consiste en inundar la red con un paquete de control Route Deletion (RD). Como alternativa para encontrar un mecanismo de eliminación menos costoso, ABR propone un mecanismo en el que las rutas son borradas después de un cierto tiempo.

5.1.2.6. Optimizaciones en protocolos reactivos

5.1.2.6.1. Uso de la caché

Al desarrollar un algoritmo de caché para un protocolo on-demand de redes inalámbricas, es fundamental la elección de la estructura de datos con que se representaría la caché. En DSR, la ruta devuelta en cada Route Reply que es recibida por el iniciador de un Route Reply representa un camino completo (una secuencia de nodos) conduciendo de aquel nodo al nodo de destino. Almacenando cada uno de estos caminos por separado se construye una path caché. O bien, se podría construir una path caché, donde cada nodo individual devuelto en la ruta de un Route Reply es añadido a un nodo unificado, que es una representación de la topología de la red.

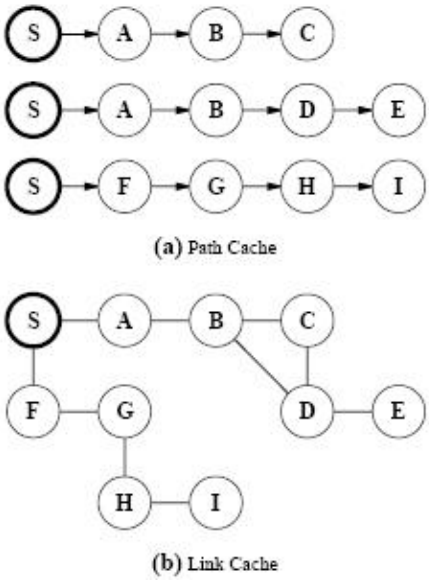


Figura 5.20. *Cachés de rutas.*

5.1.2.6.1.1. Capacidad de la caché

La capacidad de la caché es otro aspecto a tener en cuenta en el diseño de la estrategia de caché en un protocolo reactivo En una link caché lo normal es intentar almacenar todos los nodos que se descubren, sabiendo que en una red de N nodos hay un máximo de N^2 enlaces. Sin embargo, para una path caché, la capacidad de almacenamiento máxima requerida es mucho más grande, puesto que cada camino se almacena por separado.

5.1.2.6.1.2. Caché timeout

Al igual que con la capacidad de la caché, el caché timeout implica considerar una serie de alternativas en la estrategia de caché. Como las path caché tienen normalmente un mecanismo para eliminar las entradas cuando se alcanza el límite de capacidad no suelen implementar ningún tipo de timeout. En las link cachés este timeout puede ser estático o adaptativo.

5.1.2.6.1.3. Algoritmo de cachés

Aquí damos una serie de ejemplo de algoritmos de cache que se pueden utilizar con las estructuras de datos comentadas anteriormente. También se habla de Omniscient Expiration Caché que no es realmente implementable en un sistema real.

5.1.2.6.1.3.1. Path cachés

Path caches almacenan caminos completos (secuencias de enlaces) cada unos comenzando en el nodo de la caché. Ejemplos de algoritmos:

- *Path-Inf*.
- *Path-FIFO-64*.

- *Path-FIFO-32*.
- *Path-Gen-64*.
- *Path-Gen-34*.

5.1.2.6.1.3.2. Link cachés

Link caches almacenan enlaces individuales organizados en un grafo. Ejemplos de algoritmos:

- *Link-NoExp*.
- *Link-Static-5*.
- *Link-Adapt-1.25*.
- *Link-Adapt-2*.
- *Link-MaxLife.*

5.1.2.6.1.3.3. Omniscient Expiration caché

- *Link-OmniExp*.

5.1.3. Protocolos híbridos

Los protocolos de enrutamiento híbridos son una nueva generación de protocolos que son de naturaleza proactiva y reactiva. Estos protocolos se diseñan con el fin de aumentar la escalabilidad permitiendo a los nodos cercanos trabajar de forma conjunta para formar una espina dorsal que permita reducir la sobrecarga de los procesos de descubrimiento de rutas. Esto se realiza normalmente mediante un proceso de mantenimiento de rutas proactivo entre nodos cercanos, empleando una estrategia de descubrimiento de rutas reactiva para nodos lejanos. La mayoría de protocolos híbridos se basan en la creación de nodos. Los nodos pueden estar agrupados en estructuras como árboles o clusters. Vamos a estudiar el funcionamiento de dos protocolos híbridos:

- Zone Routing Protocol (ZRP).
- Zone Based Hierarchical Link State.

5.1.3.1. Zone Routing Protocol (ZRP)

El protocolo ZRP combina las ventajas de los esquemas proactivos y reactivos, manteniendo un mapa topológico actualizado de una zona centrada en cada nodo. Dentro de la zona, las rutas están disponibles de forma inmediata. Para los destinos fuera de la zona, ZRP realiza un proceso de descubrimiento de ruta, el cual se puede beneficiar de la información de enrutamiento local de las zonas.

5.1.3.1.1. Motivación

El enrutamiento proactivo consume un ancho de banda excesivo para mantener la información de enrutamiento, mientras que el enrutamiento reactivo realiza procesos de route request de gran retardo. El enrutamiento reactivo realiza también floodings

inneficientes de toda la red para descubrir rutas. ZRP intenta paliar estos problemas combinando las mejores propiedades de ambos esquemas.

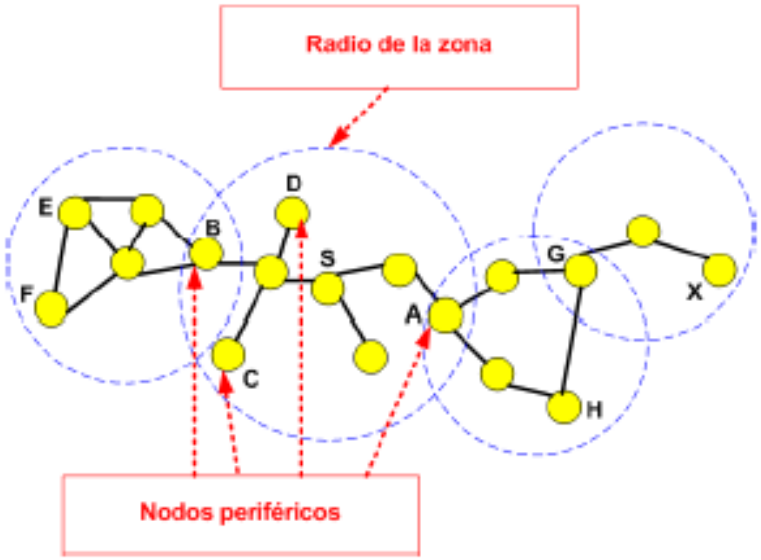


Figura 5.21. *ZRP.*

En una red ad hoc se puede considerar que la mayor parte del tráfico está dirigido a nodos cercanos. Por tanto, ZRP reduce el ámbito proactivo a zonas centradas en cada nodo. En una zona limitada, el mantenimiento de la información de enrutamiento es sencillo. De esta forma, la cantidad de información de enrutamiento que no llega a usarse se minimiza. Aún así, los nodos más lejanos pueden ser alcanzados mediante un enrutamiento reactivo. Puesto que todos los almacenan información local proactivamente, el proceso route request se puede realizar de forma más eficiente que preguntando a todos los nodos de la red. A pesar del uso de zonas, ZRP tiene una visión plana de la red. De esta forma, la sobrecarga producida por el proceso de organización asociado con los protocolos jerárquicos puede ser evitada. Los protocolos de enrutamiento jerárquicos dependen de la asignación estratégica de gateways, tales que cada nodo pueda acceder a todos los niveles, especialmente al nivel más alto. Los nodos que pertenecen a diferentes subredes deben enviar primero sus comunicaciones a una subred común para ambos nodos. Esto puede congestionar partes de la red. ZRP puede ser clasificado como un protocolo plano por el solapamiento de las zonas. De ahí que se puedan detectar rutas óptimas y la congestión de la red puede ser reducida. El comportamiento de ZRP es adaptativo y depende de la configuración de la red y el comportamiento de los usuarios.

5.1.3.1.2. Arquitectura

El Zone Routing Protocol, como implica su nombre, se basa en el concepto de zonas. Una zona de enrutamiento se define para cada nodo de forma separada, y las zonas de nodos vecinos se solapan. La zona de enrutamiento tiene un radio ρ expresado en h saltos. De esta manera, la zona incluye los nodos que distan del nodo en cuestión un máximo de h saltos. Un ejemplo de zona de enrutamiento se muestra en la Figura , donde la zona de enrutamiento de S incluye los nodos A – I, pero no K. En las

ilustraciones, el radio está marcado como un círculo alrededor del nodo en cuestión. Como vemos, la zona está definida en término de saltos, no como una distancia física.

Los nodos de una zona están divididos en nodos periféricos y nodos inteiores. Los nodos periféricos son aquellos cuya distancia mínima al nodo central es exactamente igual al radio ρ. Los nodos cuya distancia mínima es menor que ρ son nodos interiores. En la Figura , los nodos A – F son nodos inteiores, los nodos G – J son nodos periféricos y K está fuera de la zona de enrutamiento. Vemos también que H puede ser alcanzado mediante dos caminos, uno con una longitud de dos saltos y otro con una longitud de tres saltos. Sin embargo, el nodo está dentro de al zona, porque el camino más corto es menor o igual que el radio de la zona. El número de nodos en la zona de enrutamiento puede ser regulado ajustando la potencia de transmisión de los nodos. Bajando la potencia se reduce el número de nodos que son accesibles

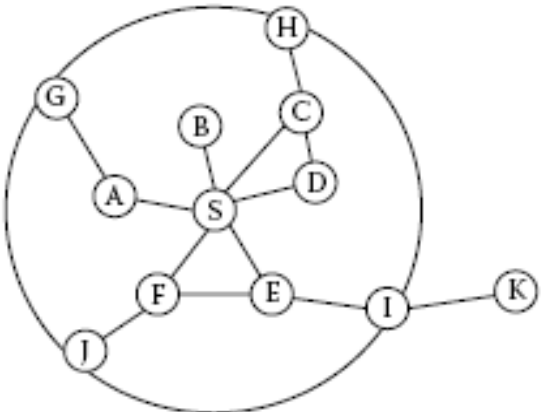


Figura5.22 *.Ejemplo de zona de enrutamiento con ρ = 2.*

directamente. El número de vecinos debería ser el suficiente para proporcionar una accesibilidad y una redundancia adecuadas. En cambio, una covertura mayor trae consigo zonas con muchos miembros y el tráfico de actualización puede resultar excesivo. Es más, el aumento de la covertura eleva la probabilidad de contención local.

ZRP se refiere al mecanismo de enrutmamiento proactivo local como el Intrazone Routing Protocol (IARP). El mecanismo de enrutamiento reactivo global es conocido como el Interzone Routing Protocol (IERP). IERP e IARP no son protocolos de enrutamiento específicos. IARP pertenece a la familia de los protocolos de enrutamiento preactivos de estado enlace con profundidad limitada. IARP mantiene información de enrutamiento para nodos que están dentro de la zona de enrutamiento del nodo. IERP es de la familia de los protocolos reactivos que ofrencen servicios de descubrimiento de ruta mantenimiento de ruta basados en la conectividad local monitorizada por IARP.

El hecho de que la topología de la zona local sea conocida, puede ser usado para reducir el tráfico cuando se necesite un descubrimiento de ruta. En vez de hacer broadcasting con los paquetes, ZRP usa el concepto de “bordercasting”. Bordercasting emplea la información de topología propocionada por IARP para dirigir los Quero request al límite de la zona. El servicio de envío de paquetes bordercast lo proporciona el Bordercast Resolution Protocol (BRP). BRP usa un mapa de una zona de

enrutamiento extendida para construir los árboles para el vío de los paquetes Quero. Alternativamente, emplea enrutamiento basado en la fuente dentro de la zona norma.

Para detectar nuevos nodos vecinos y fallos de enlace, ZRP delega en un Neighbor Discovey Protocol (NDP) proporcionado por la capa MAC. NDP transmite señales HELLO en intervalos regulares. Tras recibir una señal HELLO, la tabla de vecinos se actualiza. Los vecinos para los que no se ha recibido una señal durante un cierto período de tiempo son eliminados de la tabla. Si la capa MAC no incluye un NDP, la funcionalidad debe ser proporcionada por IARP. L relación entre los componentes se ilustra en la Figura . NDP genera actualizaciones de ruta, que son notificadas a IARP cuando la tabla de vecinos es actualizada. IERP usa la tabla de rutas de IARP para responder a las route queries. IERP envía queries con BRP. BRP usa la tabla de enrutamiento de IARP para guiar las route queries lejos de la fuente.

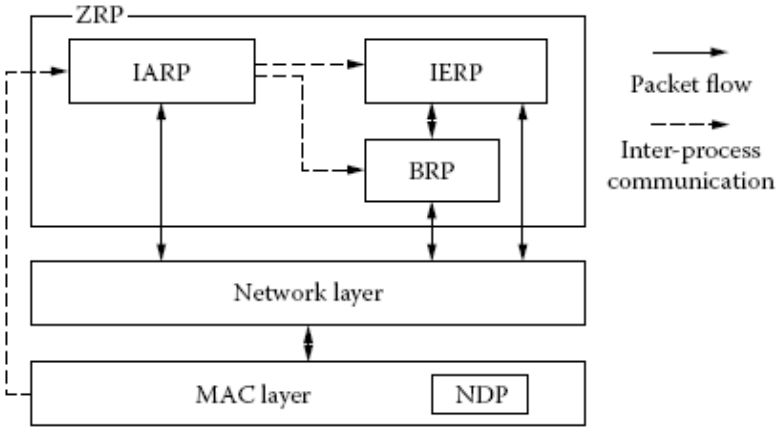


Figura5.23. *Arquitectura del Zone Routing Protocol.*

5.1.3.1.3. Enrutamiento

Un nodo que necesita enviar un paquete primero comprueba si el destinatario está dentro de su zona de enrutamiento empleando la información proporcionada por IARP. En este caso, el paquete puede ser enrutado proactivamente. El enrutamiento reactivo se emplea si se encuentra fuera de la zona.

El enrutamiento reactivo se divide en dos fases: la fase de route request y la fase de route reply. En la fase route request, la fuente envía un paquete route request a sus nodo periféricos usando BRP. Si el receptor de un paquete route request conoce al destinatario, responde devolviendo un paquete route reply al emisor. En otro caso, continua con el proceso de bordercasting del paquete. De esta forma, el paquete se difunde por toda la red. Si un nodo recibe varias copias del mismo route request, estas copias son consideradas redundantes y se descartan. Cualquier nodo que pueda proporcionar una ruta hacia el destino, envía un paquete reply. Para poder devolver un route reply a la fuente es necesario ir acumulando la información de ruta cuando el request es transmitido a lo largo de la red. La información es almacenada o bien escribiéndola en la paquete, o bien marcando el siguiente salto en cada nodo. En el primer caso los nodos que reenvían un paquete route request añaden su dirección y

métricas relevantes del nodo o el enlace al paquete. Cuando un paquete alcanza el destino, la secuencia de direcciones se invierte y copia en el paquete route reply. La secuencia es utilizar para enviar el paquete route reply de vuelta a la fuente. En el segundo caso, los nodos intermedios almacenan información de enrutamiento como la dirección del siguiente salto, el cual es utilizado para enviar el paquete route replan a la fuente. Este enfoque puede reducir el uso de recursos, ya que el tamaño de los paquetes route request y route reply se reduce. Sin embargo, los nodos intermedios deben grabar en una tabla las direcciones del siguiente salto. En el proceso de bordercasting, el nodo que realiza el proceso envía un paquete route request a cada uno de sus nodos periféricos. Este tipo de envío uno-a-muchos puede ser implementado como un envío multicasting para reducir el uso de recursos.Un esquema consiste en que la fuente calcule el árbol multicast y añada instrucciones de enrutamiento al paquete. Se le conoce como Root.Directed Bordercasting (RDB). Otra posibilidad es reconstruir el árbol en cada nodo, de forma que se puedan omitir las instrucciones de enrutamiento. Esto requiere que todos los nodos interiores conozcan la topología vista por el nodo central. Por lo tanto, los nodos mantienen una zona de enrutamiento extendida de radio 2(r - 1) saltos. Fijémonos que en este caso, el nodo periférico al que es enviado el paquete route request está a una distancia r. A este esquema se le llama Distributed Bordercasting (DB). El radio de la zona es una propiedad importante para el rendimiento de ZRP. Si se usa un radio de zona de un salto, el enrutamiento es puramente reactivo, y el bordercasting degenera en una búsqueda por flood. Si el radio se aproxima a infinito, el enrutamiento es proactivo. La selección del radio es un compromiso entre la eficiencia del enrutamiento proactivo y el incremento del tráfico para mantener la vista de la zona.

5.1.3.1.4. Mantenimiento de ruta

El mantenimiento de ruta es especialmente necesario en las redes ad hoc, donde los enlaces se rompen y se establecen debido a que los nodos se mueven unos respecto a otros y a su rango de cobertura limitado. En los protocolos puramente reactivos, las rutas que contienen errores por rupturas de enlaces generan nuevos procesos de descubrimiento de rutas o de reparación de rutas. Hasta que se encuentre una nueva ruta, los paquetes son retenidos o eliminados.

En ZRP el conocimiento de la topología local puede ser usado para el mantenimiento de rutas. Los fallos de ruta y los segmentos subóptimos de las rutas dentro de una zona pueden ser evitados. Los paquetes entrantes pueden ser dirigidos alrededor del enlace roto a través de un camino multihop activo. De forma similar, la topología puede usarse para recortar rutas, por ejemplo, cuando dos nodos entran dentro de sus respectivas zonas de cobertura. Para los paquetes dirigidos por la fuente, un nodo transmisor puede puede calcular la ruta más cercana al destino. A veces, segmentos multihop pueden ser sustituidos por un single hop. Si se usa la estrateia del siguiente salto, los nodos pueden tomar decisones óptimas loclamente para seleccionar el camino mñas corto.

5.1.3.1.5. Mecanismo de control de consultas (queries)

El bordercasting puede ser más eficiente que el flooding, porque los paquete route request sólo son enviados a los nodos periféricos y de esta forma, sólo atraviesan los enlaces necesarios. Se puede mejorar la eficiencia de forma notable empleando

técnicas multicast. En este caso, sólo se envía un paquete a un enlace, aunque varios nodos periféricos puedan residir tras ese enlace. Sin embargo, las zonas de enrutamiento de los nodos vecinos se solapan entre sí, cada nodo debe enviar route requests bastantes veces, lo que resulta en más tráfico que en el flooding. Cuando un nodo bordercast una consulta (query) se cubre la zona de enrutamiento completamente. Los posteriores mensajes que entren en la zona se consideran redundantes y provocan la pérdida de capacidad de transmisión. El exceso de trñafico es el resultado del retonro de queries para cubrir zonas, en vez de para cubrir nodos como en flooding. Para solucionar este problema, ZRP necesita mecanismos de control de consultas, los cuales pueden dirigir las solicitudes lejos de las zonas ya cubiertas y bloquear los paquetes dirigidos a nodos periféricos en regiones ya cubiertas por la consulta.

ZRP usa tres tipos de mecanismos de control de consultas: detección de consultas, teminación temprana y retraso aleatorio del procesamiento de consultas. La detección de consultas almacena en una caché las consultas ya transmitidas por los nodos. Con la terminación temprana, esta información se usa para podar el bordercastin ga nodos ya cubiertos por la consulta.

5.1.3.1.6. Detección de consultas

Cuando se realiza un bordercas, sólo el nodo central es consciente de que se está cubriendo la zona de enrutamiento con la consulta. Cuando el nodo periférico continúa el proceso de consulta haciendo bordercasting a sus nodos periféricos, la consulta puede ser transmitida a través de los mismos nodos de nuevo. Ilustrándolo con un ejemplo, vemos en la Figura que el nodo S bordercast una consulta a sus nodos periféricos F – J. Como el nodo J continúa con el bordercasting a los nodos C, S y E es redundante, porque estos nodos ya han sido cubiertos por la consulta previa.

Para prevenir que vuelvan a aparecer als consultas en regiones ya cubiertas, los nodos deben detectar la actividad local de transmisión de consultas. BRP proporciona dos métodos de detección de consultas: QD1 y QD2. En el primero, los nodos que transmiten la consulta, son capaces de detectar la consulta. E el segundo, en las redes con un solo canal, e sposible escuchar el tráfico de otros nodos dentro del radio de cobertura. De esta forma, es posible conocer el tráfico de otros nodos dentro de la zona.QD2 puede ser implementado también usando broadcast IP para enviar las consultas de ruta. Otra alternativa es usar unicast si las capas MAC e IP operan en modo promiscuo.

En el ejemplo, todos los nodos excepto B transmiten la consulta de S. Por tanto, ellos usan QD1. El nodo B no pertenece al árbol bordercast, pero es capaz de escuchar la transmisión de la consulta usando QD2. S in embargo, el nodo K no escucha el mensaje, y no es, por tanto, consciente de que la zona del nodo S está cubierta. Se emple una tabla de detección de consultas para almacenar las consultas. Para cada entrada, la caché contine la dirección del nodo fuente y el ID de la consulta. El par dirección – ID es

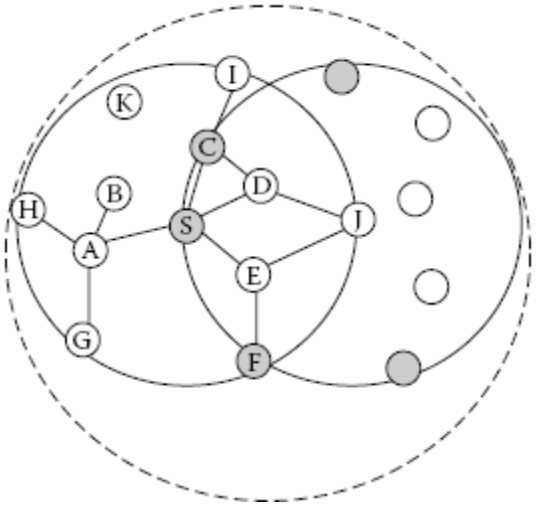


Figura 5.24 . *La zona de enrutamiento extendida de E.*

suficiente para identificar inequívocamente todas las consultas de la red. La caché puede contener otra información dependiendo del esquema de detección de consultas. En particular, la dirección del nodo que ha realizado el bordercast de una consulta puede ser importante.

5.1.3.1.7. Terminación Temprana

Con terminación temprana (ET), un nodo puede prevenir que un route request entre en una zona ya cubierta. La terminación temprana combina información obtenida a través de la detección de consultas con el conocimiento de la topología local para podar los caminos a nodos periféricos que entran en zonas ya cubiertas. Estas zonas consisten en los nodos interiores de nodos que ya han realizado un bordercasted de la consulta. Un nodo puede también podar un nodo periférico si ya ha transmitido una consulta a dicho nodo. La terminación temprana requiere una extensión de la información topológica fuera de la zona de enrutamiento del nodo. La información es necesaria para reconstruir el árbol bordercast de los demás nodos dentro de su zona de enrutamiento. La zona de enrutamiento extendida tiene un radio 2 (r - 1). En el caso del root-directed bordercasting (RDB), se puede utilizar la información de la topología de la zona de enrutamiento standard. En el ejemplo previo, el nodo E puede usar la información en su tabla de detección de consultas para podar la consulta que el nodo J envía a su nodo periféico F. El nodo E tiene una zona de enrutamiento extendida con radio 2 (r - 1) = 3 como muestra el círculo discontinuo de la Figura .

5.1.3.1.8. Retraso Aleatorio en el procesamiento de consultas

Cuando un nodo envía un route request, almacena algún tiempo la consulta que se va a transmitir a través del árbol bordercast para poder ser detectada por los

mecanismos de detección. Durante este tiempo, otro nodo podría propagar la misma route request. Esto puede ser un problema cuando varios nodos cercanos reciben y reenvían aproximadamente al mismo tiempo. Para reducir la probabilidad de recibir a misma route request de varios nodos, se puede utilizar un Random Quero-Processing Delay (RQPD). Cada nodo bordercasting espera un tiempo aleatorio antes de la construcción del árbol bordercast y el de terminación temprana. Durante este tiempo, el nodo en espera puede detectar otros nodos bordercasting y podar el árbol bordercast. Este retardo puede ser combinado con el jitter de la pretransmisión utilizado por algunos protocolos de descubrimiento de ruta.

Suponemos que en la Figura 5.24, los nodos C y S reciben una consulta. El nodo C planifica un bordercast a su nodo periférico E, y nodo S a su nodo periférico F. Sin RQPD, ambos nodos realizarían el broadcast simultáneamente, y a paritr de entonces detectan el mensaje de su nodo vecino. Con RQPD, el nodo C puede detectar el envío de una consulta por un nodo S durante el período de retraso, y podar la rama hacia E.

5.1.3.1.9. Caching

El almacenamiento en caché es una técnica usada para reducir el tráfico de control. Los guardan en caché las rutas activas, y usando esta caché la frecuencia de ejecución de descubrimientos de rutas se reduce. Los cambios en la topología de la red, como la ruptura de enlaces, se compensan con la reparación local de caminos. Un nuevo camino sustituye entonces al camino entre los finales del enlace roto, y se manda un mensaje path update a los extremos del camino. Puesto que la reparación reduce la calidad de las rutas, los extremos podrían decidir iniciar un nuevo descubrimiento de ruta después de un número de reparaciones locales.

5.1.3.2. Zone Based Hierarchical Link State (ZHLS)

El protocolo Zone Based Hierarchical LSR Protocol (ZHLS) es un protocolo de enrutamiento “peer-to-peer” jerárquico que incorpora información de localización en un nuevo esquema “peer-to-peer” jerárquico. La red se divide en zonas no superpuestas. Agregando los nodos a las zonas se ocultan los detalles topológicos de la red. Inicialmente, cada nodo conoce su posición y el ID de la zona a través del GPS. Una vez se ha establecido la red, cada nodo conoce la topología a más bajo nivel (nivel de nodo) de las conexiones dentro de su zona, y la topología de alto nivel (nivel de zona) de la conectividad de toda la red. Se envía un paquete especificando la dirección jerárquica (ID del nodo e ID de la zona) del nodo destino en la cabecera del paquete. A diferencia de otros protocolos. A diferencia de otros protocolos jerárquicos, no utiliza clusterheads. La información topológica de alto nivel es distribuida a todos los nodos. Esta característica “peer-to-peer” evita cuelos de botella del tráfico, previene los fallos puntuales y simplica el control del movimiento. De forma similar a ZRP, ZHLS es un esquema híbrido. Es proactivo si el destino se encuentra en la misma zona que la fuente. En caso contrario, porque es necesaria la búsqueda de su localización para encontrar el ID de la zona a la que pertenece. Sin embargo, a diferencia de ZRP, ZHLS requiere GPS y mantener una estructura jerárquica de alto nivel para el enrutamiento interzone. La búsqueda de su localización se realiza mediante un envío unicast de un request location a cada zona. El enrutamiento se realiza indicando el ID de la zona y el ID del nodo destino, en vez de especificar una lista de nodos intermedios entre la fuente y el

destino. La ruptura de un enlace intermedio no ocasiona otra búsqueda. Puesto que la red consiste en zonas no superpuestas, se pueden reutilizar las rutas frecuentemente.

5.1.3.2.1. Mapa zonal

La red se divide en zonas. Un nodo conoce su localización física gracias a técnicas geolocalizadoras como un GPS; luego, puede determinar el ID de su zona mapeando su localización física a un mapa zonal, el cual ha debido ser elaborado durante el diseño del escenario. El tamaño de la zona depende de algunos factores como la movilidad de los nodos, la densidad de la red, la potencia de transmisión, y las carácterísticas de propagación. El particionamiento de la red puede estar basado simplemente en datos geográficos o en la propagación de la señal de radio. El particionamiento geográfico es mucho más sencillo y no requiere tomar medidas de las características de la propagación d ela señal de radio, mientras que el particionamiento por propagación de señal de radio es más adecuado para una frecuente reutilización. Este método es preferible si se pueden realizar las medidas necesarias durante el proceso de diseño del escenario. Sin embargo, para muchas aplicaciones como las operaciones de rescate en un desastre, comunicaciones tácticas militares, etc, no se pueden tomar estas medidas. En estos casos, se utiliza una partición geográfica.

5.1.3.2.2. Estructura jerárquica de la zona

Se definen dos niveles de topología en ZHLS: la topología a nivel de nodo y la topología a nivel de zona. Si dos nodos cualesquiera se encuentran dentro de su rango de comunicación, existe un enlace físico. La topología a nivel de nodo proporciona información sobre como están los nodos conectados a través de esos enlaces físicos.

Por ejemplo, en la Figura 5.26, si el nodo a quiere enviar un paquete de datos a un nodo i, los datos deben pasar a través de a – b – c – f. Si hay al menos un enlace físico uniendo dos zonas, existe un enlace virtual. La topología a nivel de zona, indica como los nodos están conectados por estos enlaces virtuales. Por ejemplo, en la Figura , los enlaces virtuales entre la zona 4 y la 3 son 4 – 1 – 3.

Vamos a ver ahora como un nodo usa la topología a nivel de nodo para enlutar un paquete dentro de la zona y como emplea la topología a nivel de zona para enlutar un paquete entre zonas. Para facilitar esta protocolo LSR jerárquico, cada nodo recibe dos tipos de paquetes de estado de enlace (LSPs): LSPs de nodo y LSP de zona. El LSP de nodo de un nodo particular contiene una lista de sus vecinos conectados, y se propaga localmente dentro de la zona. Los LSP de zona contienen una lista de sus zonas conectadas y son propagados localmente a través de la red.

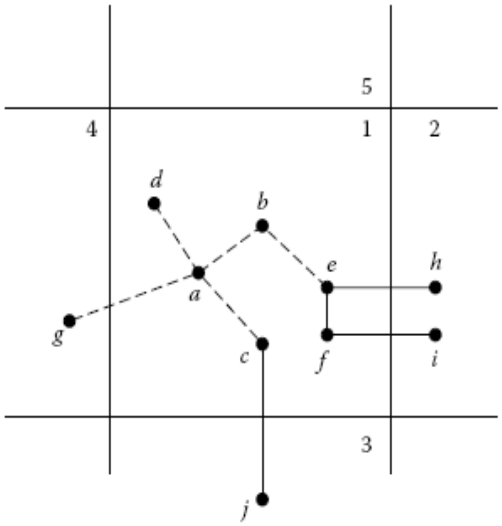


Figura 5.25 *. Topología a nivel de nodo.*

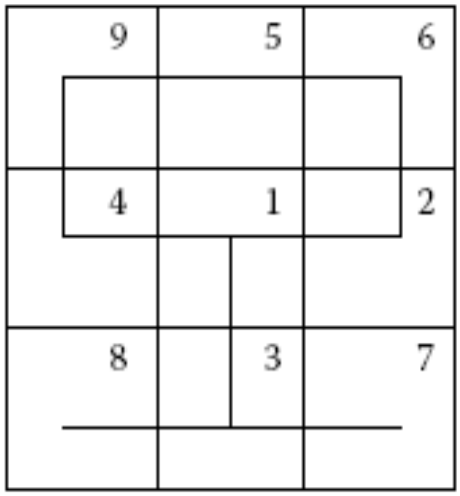


Figura 5.26*. Topología a nivel de zona.*

5.1.4. Protocolos basados en la localización

5.1.4.1. Location-Aided Routing (LAR)

Los protocolos LAR usan la información de localización (la cual puede estar desactualizada cuando se usa) para reducir el espacio de búsqueda para la ruta deseada. Limitar el espacio de búsqueda conlleva una reducción del número de mensajes de descubrimiento de ruta.

5.1.4.2. Descubrimiento de Ruta usando Flooding

La posibilidad de usar información de localización para mejorar el rendimiento de los protocolos de enrutamiento en MANET está siendo estudiada. En la Figura se muestra como un protocolo de descubrimiento de ruta basado en flooding puede ser mejorado. El algoritmo de descubrimiento de ruta usando flooding se describe a continuación. Cuando un nodo S necesita encontrar una ruta al nodo D, el nodo S broadcast un mensaje route request a todos sus vecinos; En lo sucesivo llamaremos emisor al nodo S y destino al nodo D. Un nodo X que recibe un mensaje route request, compara el destino deseado con su propio identificador. Si coincide, quiere decir que ha recibido una solicitud de ruta para sí mismo. En caso contrario, el nodo X broadcast el mensaje route request a sus vecinos

5.2. Enrutamiento Multicast

La tecnología *multicast* representa un servicio de red en el cual un único flujo de datos, proveniente de una determinada fuente, se puede enviar simultáneamente a diversos receptores interesados. Cabe a la infraestructura de red transportar este flujo de datos, replicándolo cuando sea necesario, para todos los receptores que registren interés en recibir estos datos.

En redes TCP/IP, estos receptores son representados por una dirección de grupo o dirección *multicast*. Esta dirección de grupo corresponde a una dirección IP que pertenece a la antigua clase D, es decir, en la franja entre 224.0.0.0 y 239.255.255.255. Cada fuente envía paquetes hacia una dirección de grupo (por ejemplo: 233.7.124.1), en el cual estarán asociados diversos receptores. Estos receptores, a su vez se pueden vincular y desvincular en forma dinámica. Cabe a los dispositivos de la red y en particular a los enrutadores, determinar cuáles de sus interfaces poseen receptores interesados en un grupo *multicast* y cuáles deberán recibir una copia de los paquetes enviados para ese grupo.

El *multicast* está orientado hacia aplicaciones del tipo "uno para muchos" y "muchos para muchos". En estos casos, presenta claras ventajas cuando se lo compara con los mecanismos de transmisión *unicast* y *broadcast*. En *unicast*, es necesario que la fuente replique varios flujos de datos idénticos con el objeto de transmitirlos a cada uno de los receptores, generando desperdicio de banda. Por otro lado, el sistema *broadcast* envía los datos a toda la red de forma indiscriminada. Esto también da como resultado el desperdicio de recursos, pues implica en transporte de datos para todas las estaciones de la red, aunque el número de receptores deseosos de que ese contenido sea reducido. Con *multicast*, la fuente de tránsito envía una única copia de los paquetes hacia una dirección de grupo *multicast*. La infraestructura de red replica estos paquetes de forma inteligente, encaminando los datos de acuerdo con la topología de receptores interesados en esa información.

Entre las diversas aplicaciones que pueden obtener ganancias con el uso de *multicast* están: videoconferencia; aprendizaje a distancia; distribución de software, noticias e informaciones de mercado; conciertos al vivo; actualización de bases de datos; juegos distribuidos; procesamiento competidor; simulacros distribuidos etc...

Es las MANET, donde los recursos son limitados, el uso de multicast en las aplicaciones “uno para muchos” es más importante si cabe que en las redes tradicionales, debido a la reducción de la cantidad d einformación enviada. Sin embargo, es más difícil crear un árbol de rutas que lleve los datos a los diferentes destinos y mantener todas las rutas puede ser costoso. Además, hay que llegar a un compromiso entre la calidad de las rutas y la cantidad de información redundante que permitimos. Vamos a explicar este punto más detenidamente. Ya hemos comentado anteriormente que las MANETs son redes multi-hop, donde todos los nodos actúan como routers y terminales. Por lo tanto, para realizar un envío, el mensaje debe atravesar varios nodos intermedios hasta llegar a su destino. En el caso del envío multicast tendremos un mensaje pero varios destinatarios. En el caso más simple, con dos destinatarios adyacentes, el envío consumiría prácticamente los mismos recursos que un envío multicast, pues bastaría seguir la ruta común a los dos y la última retransmisión llevaría el mensaje a su destino. En envíos más complejos, con más nodos y con situaciones aleatorias, no se podrá encontrar una ruta común que alcance a todos los destinatarios, pero será interesante que el mensaje no se divida hasta lo más tarde posible. Sin embargo, al intentar fusionar las rutas, necesariamente nos estaremos desviando de la ruta óptimas, con lo que el tiempo de envío crecerá. Por tanto, deberemos llegar a un compromiso entre estos dos aspectos, sabiendo que fusionar rutas provocará un mayor tiempo de envío y separarlas aumentará la sobrecarga de la red (en el caso extremo, el multicast se convertirá en un conjunto de envíos unicast).

5.2.1. Algoritmos Multicast

Los algoritmos propuestos a día de hoy para realizar envíos multixcast en redes ad hoc se dividen en dos grupos:

- **Aproximación de árbol.** Crea un árbol con el origen en la raíz y los destinos en las hojas. Usa una métrica, como el número de saltos. AMRIS.
- **Aproximación de malla (grafo).** Se adapta mejor a las redes más dinámicas debido a la redundancia de ésta aproximación. Existe más de un camino entre origen y destino, así si un camino falla, existen otras posibilidades. ODMRP.

Vamos a comentar los ejemplos más destacados.

5.2.1.1. On-Demand Multicast Routing Protocol (ODMRP)

El protocolo *On-Demand Multicast Routing Protocol*, (ODMRP) descrito por Lee, Gerla y Chiang es un protocolo de encaminamiento multicast para redes ad hoc con nodos móviles. Utiliza un esquema multicast en malla en lugar del clásico esquema en árbol. Dentro de la malla se utiliza inundación restringida. Un conjunto reducido de nodos (*Forwarding Group*) se encarga de las retransmisiones. Emplea procedimientos típicos de los protocolos reactivos para construir rutas y mantener los grupos multicast.

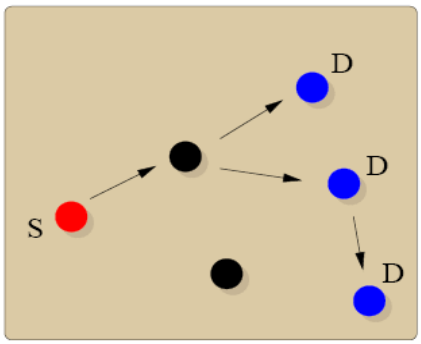


Figura5.27. *Esquema de un algoritmo multicast.*

ODMRP se adapta bien a redes Ad-Hoc inalámbricas con nodos móviles donde el ancho de banda es limitado y la topología cambia rápida y frecuentemente.

ODMRP construye un árbol de caminos más cortos (*Shortest Path Tree, SPT*) o malla de relays para cada grupo multicast. Cuanto mayor sea el número de receptores y fuentes, más nodos del conjunto de relays serán utilizados en distintos SPT por lo que aumenta el porcentaje de destinos alcanzados. Sin embargo esta capacidad de adaptación implica una alta sobrecarga de la red en cuanto a mensajes de control.

En ODMRP un nodo *S* fuente de un grupo multicast *G*, inunda periódicamente (normalmente cada tres segundos) la red con un mensaje de solicitud de unión a grupo (*JOIN REQUEST*). Cada nodo *R* interesado en recibir mensajes enviados al grupo *G*, contesta por el camino más corto hasta *S* con un mensaje de respuesta (*JOIN REPLY*). Antes de reenviar un mensaje JOIN REPLY, cada nodo espera cierto tiempo en previsión de que más nodos usen el mismo camino más corto hasta la fuente. De este

modo se reduce el número de mensajes JOIN REPLY circulando hacia el nodo *S*. Cuando un mensaje JOIN REQUEST llega a un receptor, este actualiza su tabla de pertenencia a grupos. Las tablas son distribuidas dentro de la malla periódicamente. De este modo, las rutas se crean o actualizan.

Cuando los nodos fuente emiten sus paquetes de datos para un grupo multicast, éstos fluyen por el grupo de nodos que compone el SPT hacia los nodos receptores que contestaron con un JOIN REPLY al mensaje JOIN REQUEST para el grupo *G*. El tiempo durante el cual las rutas se mantienen válidas es equivalente a múltiples emisiones de JOIN REQUEST. No es necesario el intercambio explícito de mensajes HELLO. Esto hace que las rutas se mantengan demasiado tiempo y en ocasiones permite la creación de múltiples rutas en paralelo, es decir, múltiples árboles solapados para el mismo grupo multicast.

5.2.1.2. CAMP

García-Luna-Aceves y Madruga describen el denominado *Core-Assisted Mesh Protocol*, (CAMP). Este protocolo construye y mantiene una malla multicast para la distribución de información para cada grupo multicast. Cada malla es un subconjunto de la topología de la red en la existe al menos un camino entre cada fuente y cada destino del grupo multicast. CAMP asegura que los caminos más cortos entre fuentes y receptores forman parte de la malla. Al igual que en CBT, se utiliza el concepto de nodo núcleo, del inglés *core*, como centro de gestión de los grupos multicast. La principal diferencia con CBT es que el uso de estos nodos no es obligatorio.

Cuando un nodo desea unirse a un grupo multicast tiene tres opciones. Enviar el mensaje a alguno de sus vecinos que ya esté unido al mismo grupo, enviarlo a un nodo core o utilizar inundación para buscar o bien un nodo core o algún miembro del grupo al que poder enviarle la solicitud de unión. Los nodos core periódicamente incluyen su dirección en los mensajes de pertenencia al grupo normales. Gracias a que los nodos que forman parte de la malla multicast, existen distintas caminos que unen los nodos de la misma malla. Con esto, se consigue un mayor índice de tolerancia a los cambios en la topología, lo que implica una menor necesidad de reconstrucción de rutas y por lo tanto menor sobrecarga.

5.2.1.3. MAODV

La extensión multicast de AODV conocida como Multicast AODV o MAODV pretende construir árboles multicast bidireccionales compartidos que conecten múltiples fuetes y destinos para cada grupo multicast. Estos árboles se mantienen mientras que existen miembros del grupo multicast conectados a alguna parte del árbol. Cada grupo multicast tiene un nodo líder, responsable de mantener el valor del número de secuencia del grupo que además es la raíz del árbol multicast. Este número es usado para asegurar la precisión de la información de encaminamiento. Un nodo se convierte en líder tras varios intentos infructuosos de unirse a un grupo multicast inundando la red con mensajes de unión al grupo.

Los nodos líderes de grupo inunda la red periódicamente anunciando su dirección y su situación de líder de grupo así como el número de secuencia del grupo. Cuando un nodo desea enviar mensajes a dicho grupo multicast para el cual no conoce ningún líder, primero intenta hacerse líder de grupo. Si no recibe respuesta él mismo se convierte en líder y comienza a emitir. Si ya conocía la identidad del líder por haber recibido previamente un mensaje de anuncio, envía los mensajes de datos directamente al líder del grupo para que este los distribuya por el árbol multicast.

Cuando un nodo desea unirse a un grupo como receptor, envía una petición inundando la red. Estas peticiones pueden ser contestadas por cualquier miembro del árbol. Las respuestas son enviadas al originador de modo que los nodos por los que pasa se convierten en nuevos miembros del árbol multicast. Esta forma de construir los árboles genera rutas demasiado largas con una mayor probabilidad de rotura y por consiguiente pérdida de paquetes.

5.2.1.4. AMRIS

Wu y Tay describieron el protocolo *A Multicast Protocol for Ad hoc Wireless Networks*, AMRIS, en. AMRIS es un protocolo multicast bajo demanda que construye árboles compartidos para soportar múltiples emisores y receptores en un mismo grupo multicast. Cada nodo tiene un identificador denominado *msm-id* dinámicamente calculado durante la fase de inicialización. El valor del *msm-id* en cada nodo debe ser mayor que el de su padre lógico en el árbol multicast. La fase de inicialización arranca cuando un nodo especial denominado *Sid* es elegido de entre los distintos emisores. El *Sid* tendrá el menor msm-id de todos los participantes en el grupo multicast y será por tanto la raíz del árbol.

El *msm-id* permite a los nodos separados del árbol por cambios en la topología, volver a unirse al árbol sin causar ciclos. Cada nodo envía periódicamente un mensaje a sus vecinos informándoles de su presencia y su *msm-id*

.

AMRIS consiste en dos fases principales. La inicialización del árbol y su mantenimiento.

La inicialización consiste en la inundación de la red con un mensaje de anuncio por parte del nodo *Sid*. Cada nodo, al recibir estos mensajes de anuncio calcula su *msm-id* sumándole un valor aleatorio que sustituye al original en el mensaje. El intercambio periódico de mensajes de presencia sirve para que cada nodo pueda determinar qué vecino es su padre en el árbol multicast. Un nodo puede recibir mensajes de anuncio procedentes de distintos vecinos. Tras comparar los valores de *msm-id* y otras métricas incluidas en la cabecera, se puede cambiar de padre si se determina que el nuevo es mejor.

Cuando un nodo desea recibir mensajes de un grupo multicast, envía un mensaje unicast de petición *JOINREQ* a cada uno de sus padres en el árbol multicast (aquellos con menor *msm-id* que él. Cuando el nodo recibe el JOINREQ responde con un JOINACK si el es miembro de la sesión multicast que se solicita. En otro caso, se envía un JOINREQPASSIVE a sus padres potenciales. Si un nodo no recibe un JOINACK o lo que recibe es un paquete JOINNAK como respuesta a su JOINREQ, entonces tiene que realizar lo que se denomina reconstrucción de la rama *Branch Reconstruction, BR*.

El proceso BR realiza una búsqueda expandida por inundación hasta el nodo que complete la unión a la sesión multicast.

Por último, es importante indicar que AMRIS detecta los enlaces desconectados mediante el envío periódico de paquetes de presencia. Cuando un nodo no reciba paquetes de anuncio de uno de sus vecinos, considera que éste se ha movido. Excepto en el caso que el vecino fuera un padre suyo por lo que el nodo debe volver a unirse al árbol enviando un paquete JOINREQ a un nuevo padre potencial. Si el JOINREQ no es respondido con un JOINACK entonces se tiene que realizar el proceso BR, anteriormente comentado. Los reenvíos de datos solo los hacen los nodos del árbol. Por tanto, los paquetes son reenviados de los padres a los hijos. Con el posible problema de que si se rompe algún enlace, los paquetes se van a perder hasta que se reconstruya el árbol de nuevo.

5.2.1.5. ADMR

El protocolo *Adaptive Demand-Driven Multicast Routing*, (ADMR), descrito por Jetcheva y Johnson no utiliza mensajes de control periódicos ni depende de ningún protocolo de nivel inferior para realizar esta u otras funciones. ADMR construye árboles multicast distintos para cada receptor y grupo multicast. Teniendo como raíz de los árboles multicast los receptores, los caminos generados desde las fuentes son los más cortos en número de saltos.

Cuando un nodo *S* desea generar tráfico para un grupo multicast *G*, inicia una inundación controlada de la red con un mensaje de descubrimiento de ruta (*ROUTE-DISCOVERY*) que incluye un contador de saltos además de la dirección de *S*. Este contador es incrementado por cada nodo que recibe y reenvía el mensaje. El reenvío sólo produce una vez pero cada vez que se recibe un mensaje de este tipo, se actualiza el estado referente al camino hacia *S* para el grupo *G* más corto. De este modo los posibles nodos relays del árbol saben cual es el camino más corto en saltos hacia la fuente.

Cuando un nodo *R* interesado en recibir datos del grupo *G* recibe un mensaje ROUTE-DISCOVERY, contesta con un mensaje de unión al grupo (*RECEIVER-JOIN*) que es enviado de vuelta hacia la fuente por el camino más corto en número de saltos. Cada nodo por el que pasa este mensaje se configura como relay para la pareja < *S, G* >. Por lo tanto, cuando *S* envíe tráfico marcado con el grupo *G*, sólo los relays de < *S, G* > reenviarán los mensajes.

Los cambios en la topología son detectados por ADMR gracias a un mecanismo de asentimiento pasivo (*passive acknowledgement*). Cada nodo actuando como relay reenvía el mensaje recibido y a continuación se queda esperando a recibir el envío del siguiente nodo de la cadena. Como todos los envíos se realizan en un medio compartido, si asumimos que los enlaces son bidireccionales, esta operación es posible. Cuando un relay no escucha el reenvío del siguiente nodo en el árbol, borra su entrada en la tabla y deja de comportarse como relay para < *S,G* >. El mecanismo para reconstruir árboles rotos consiste en una operación de redescubrimiento periódica. La periodicidad debe limitarse al mayor periodo de tiempo que el nivel de aplicación pueda soportar sin recibir datos. De esta forma se reduce al mínimo la sobrecarga que produce el redescubrimiento. Existe una especificación en la que se adapta ADMR a redes de

sensores. El protocolo es difícil de adaptar debido al exceso de memoria que requiere para almacenar el estado en los nodos.

5.2.1.6. DSR-MB

Una sencilla extensión de DSR especifica que para entornos de redes Ad-Hoc pequeñas con mucha movilidad puede ser más eficiente el flooding que la creación y mantenimiento de estructuras para el reenvío de paquetes multicast o broadcast. En este trabajo se presenta DSR-MB o Simple MBCAST, un protocolo en el cual cuando un nodo recibe por primera vez un mensaje de tipo multicast o broadcast, simplemente lo reenvía a todos sus vecinos. No se requiere que los nodos almacenen estado de ninguna clase. Se trata de una estrategia muy simple que sólo tiene utilidad en ciertos escenarios muy concretos. Por lo tanto, no se puede considerar como un protocolo para uso general de multicast.

5.2.1.7. MMRP

El protocolo *Mobile Mesh Routing Protocol*, (MMRP) es parte de una familia en la que también se incluyen MMLDP y MMBDP. Con estos tres protocolos se puede construir una red Ad-Hoc flexible y extensible. MMRP se basa en estado de enlace, es decir, en el intercambio de la información que cada nodo tiene sobre sus vecinos. En MMRP genera mensajes conteniendo información sobre sus vecinos que son enviados en un broadcast limitado. Estos mensajes, llamados LSP, se propagan un número limitado de saltos, no a toda la red. La idea es aprovechar la posible localidad espacial de las comunicaciones. El tamaño de esta área, o lo que es lo mismo, el número máximo de saltos que se puede transmitir un mensaje LSP, forman parte de los parámetros del protocolo y se deben configurar dependiendo del tipo de red.

En una red con una topología muy cambiante, la probabilidad éxito de una comunicación multi-salto decrece al incrementarse la distancia. Por lo tanto, llevar los mensajes LSP a gran distancia no tiene utilidad. Esto reduce en parte la sobrecarga, especialmente cuando la movilidad es muy alta.

5.2.1.8. MMARP

Propuesto por Ruiz *et al*, el protocolo *Multicast Routing for MANET Extensions to IP Access Networks*, (MMARP) está especialmente diseñado para redes Ad-Hoc móviles.

Es totalmente compatible con el standard multicast sobre IP y permite que nodos móviles con soporte de IP standard puedan formar parte de comunicaciones multicast ya que MMARP utiliza IGMP como medio de control para interoperar no sólo con los nodos fijos y móviles. La comunicación con los nodos fijos que actúan como puentes de conexión para la red móvil, se realiza a través de los nodos móviles situados a un salto de ellos. Estos nodos son denominados MIGs del inglés *Multicast Internet Gateway*.

En MMARP se utiliza una malla de distribución multicast similar a a la de ODMRP lo que le ofrece una buena protección contra movilidad. Para la creación y

mantenimiento de dicha malla, MMARP usa un enfoque híbrido. Las rutas multicast entre nodos móviles se establecen bajo demanda, al estilo de los algoritmos reactivos, sin embargo, cuando la fuente del tráfico multicast está en un nodo de la red fija, se utiliza la construcción proactiva de rutas. En concreto, la parte reactiva consiste en la utilización de la inundación para anunciar, mediante el uso de mensajes de tipo *MMARP SOURCE*, la existencia de una nueva fuente.

Los nodos receptores, contestan los mensajes de anuncio con mensajes de respuesta *MMARP JOIN* creando de este modo la ruta hasta el origen. Al superponerse los caminos creados por diferentes nodos receptores se crea un multicast Shortest Path Tree (SPT), utilizado posteriormente para el envío de los datos. Por otro lado, la parte proactiva del protocolo consiste simplemente en el anuncio periódico de los MIGs mediante inundación. De este modo, los nodos móviles saben cual es su puente de conexión a la red fija.

Los mismos autores proponen en una técnica para reducir el número de nodos retransmisores (*Forwarding Nodes*) en aquellos algoritmos que usen malla multicast en lugar de un árbol. El objetivo de reducir el número de estos nodos es construir árboles cuyo coste se aproxime al mínimo posible. Inspirándose en los algoritmos epidémicos, añaden a los mensajes de creación de rutas un contador denominado FNCount cuyo valor es incrementado por cada nodo que reenvía el mensaje si no es ya un miembro del grupo de nodos retransmisores. El contador se pone a cero cuando es enviado por un nodo que, al mismo tiempo, es un receptor. De este modo, los posibles receptores pueden elegir las rutas hacia la fuente, dando preferencia a aquellas con un menor valor de FNCount. Los autores concluyen que, tras aplicar la técnica definida al algoritmo ODMRP, el porcentaje de paquetes entregados se mantiene similar y, al mismo tiempo, la reducción en el número de nodos necesarios para mantener la malla multicast, es de entre un 25% y un 50 %.

5.3. Broadcast en redes redes inalámbricas multihop

Hemos comentado anteriormente que los protocolos reactivos necesitan inundar mensajes de control que van destinados a toda la red (broadcast), pero esto también

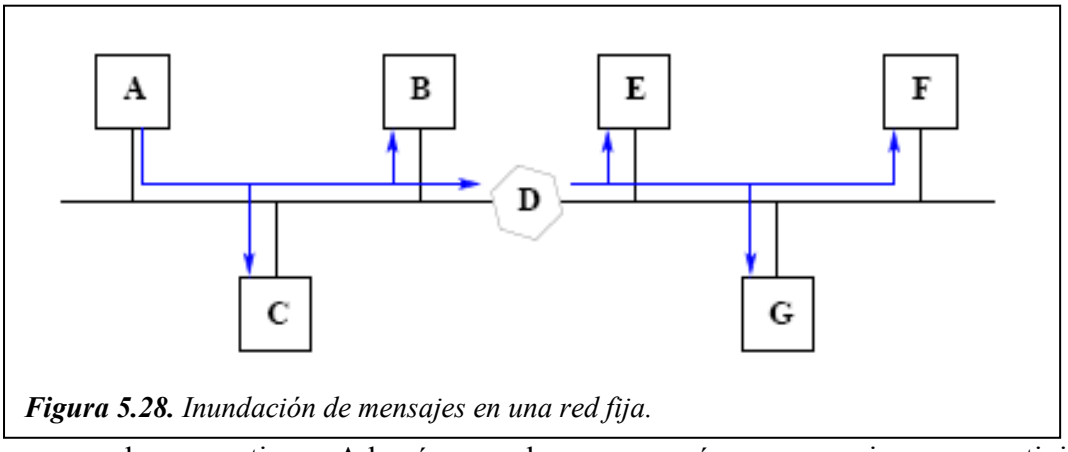


Figura 5.28. *Inundación de mensajes en una red fija.*

ocurre en los proactivos. Además cuando veamos cómo proporcionar conectividad global a las manet tendremos otro ejemplo de que la inundación de mensajes del protocolo de enrutamiento es necesaria para el correcto funcionamiento del mismo.

La inundación tradicional de mensajes broadcast consiste en retransmitir el mensaje por todas las interfaces del host salvo por la que se ha recibido. En la Figura vemos cómo el host A inunda un mensaje en una red fija. Sólo es necesario poner dos mensajes en la red, el original que envía A y el que retransmite D. Sin embargo en las MANETs el escenario se complica. En la Figura 5.29 vemos la red ad hoc equivalente a la fija de la Figura 5.28. Ahora tenemos que reenviar los mensajes por la misma interfaz por la que nos llegan debido a la naturaleza multi-salto (multi-hop) de las manet.

También es necesario que los nodos almacenen temporalmente los mensajes que retransmiten (o al menos un identificador de los mismos) para evitar enviar mensajes duplicados. Aún teniendo esto en cuenta, ahora se ponen siete mensajes en la red (Figura 5.30). Esto supone una gran sobrecarga para la red ad hoc que como sabemos está más limitada en ancho de banda que su contrapartida en redes fijas. Como además el medio es compartido por todos los nodos el efecto negativo se multiplica: los nodos deben competir entre sí para adquirir el medio introduciendo un mayor retardo.

El problema que hemos descrito se acentúa conforme se aumenta el número de nodos y la densidad de la red. Como además es un mecanismo básico para los protocolos de enrutamiento, es importante buscar mejoras. Existen muchas optimizaciones, pero nosotros estudiaremos sólo las dos que más nos interesan, la búsqueda por expansión del anillo (expanding-ring search) y los MPR (multi point relays). Ambas técnicas son utilizadas por los protocolos de enrutamiento que ya hemos analizado.

5.3.1. Expanding-Ring Search

Esta técnica no es original de las manet, ya que su uso es muy antiguo incluso en redes fijas. Tiene utilidad cuando se quiere descubrir algo que no se sabe donde está pero se tiene una cierta esperanza en que se encuentre cerca. Entonces lo que se hace es inundar un mensaje broadcast que indica qué es lo que se está buscando, pero se limita la inundación a una zona reducida. Si no hay respuesta se amplía un poco la zona de inundación y se repite el proceso. Se opera así hasta que hay una respuesta del destino (o destinos) o se asume que no es alcanzable.

Como vemos esta técnica de inundación consiste precisamente en no inundar la red, sino sólo parte de ella. ¿Por qué es esto útil en las MANET? Porque los protocolos reactivos operan precisamente así cuando quieren descubrir una ruta: como no conocen donde está el destino tienen que inundar un mensaje para encontrarlo. Y cuando veamos las propuestas de conectividad a Internet nos daremos cuenta de que también es habitual emplear esta técnica para descubrir un gateway que de acceso a Internet.

Aplicar este mecanismo en redes IP es tan secillo como ir variando el campo TTL de IPv4 o Hop Limit de IPv6, de forma que empiece con un valor pequeño y vaya aumentando si no se obtiene respuesta.

5.3.2. Multi Point Relays

La técnica de los MPR está diseñada para reducir el número de retransmisiones redundantes que tienen lugar por parte de los nodos de la red ad hoc. Antes veíaamos cómo se realizaban siete retransmisiones en una MANET cuando sólo eran necesarias dos de ellas: una del nodo A y otra del nodo D (Figura 5.30). Por ejemplo, el nodo B retransmite el mensaje porque no tiene la información necesaria para saber que todos sus vecinos son también vecinos de A, y por tanto reenviar el mensaje no tiene sentido.

El mecanismo MPR se basa en reducir el conjunto de nodos de la red que van a retransmitir los mensajes broadcast, de tal forma que el mensaje llegue a todos los nodos de la manet (no como ocurría en el expanding-ring search) con el menor número de retransmisiones. Cada nodo va a calcular cuál es ese conjunto mínimo de vecinos que deben retransmitir sus mensajes de control, y sólo ellos efectuarán las retransmisiones. Dicho conjunto mínimo es lo que se conoce con el nombre de multi point relays, y se calcula averiguando el menor número de vecinos que se necesitan para alcanzar a todos los nodos que se encuentran a dos saltos de distancia.

Para averiguar el conjunto de MPRs es necesario intercambiar previamente cierta información acerca de la topología local de la red.

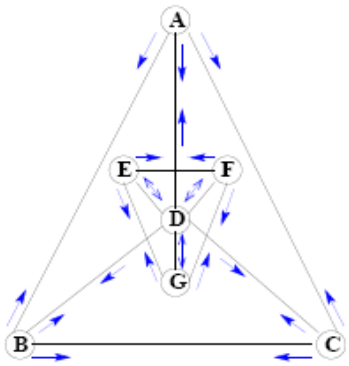


Figura 5.29 . *Inundación de mensajes en una MANET.*

En el ejemplo de la Figura 5.30 si el nodo emisor (el nodo A) y su relay (el nodo D) disponen de toda la información necesaria y actualizada, se consigue el objetivo empleando sólo dos transmisiones, es decir, se alcanza el óptimo. En general esto no es as, pero la técnica de los MPR propicia un número limitado de transmisiones.
í

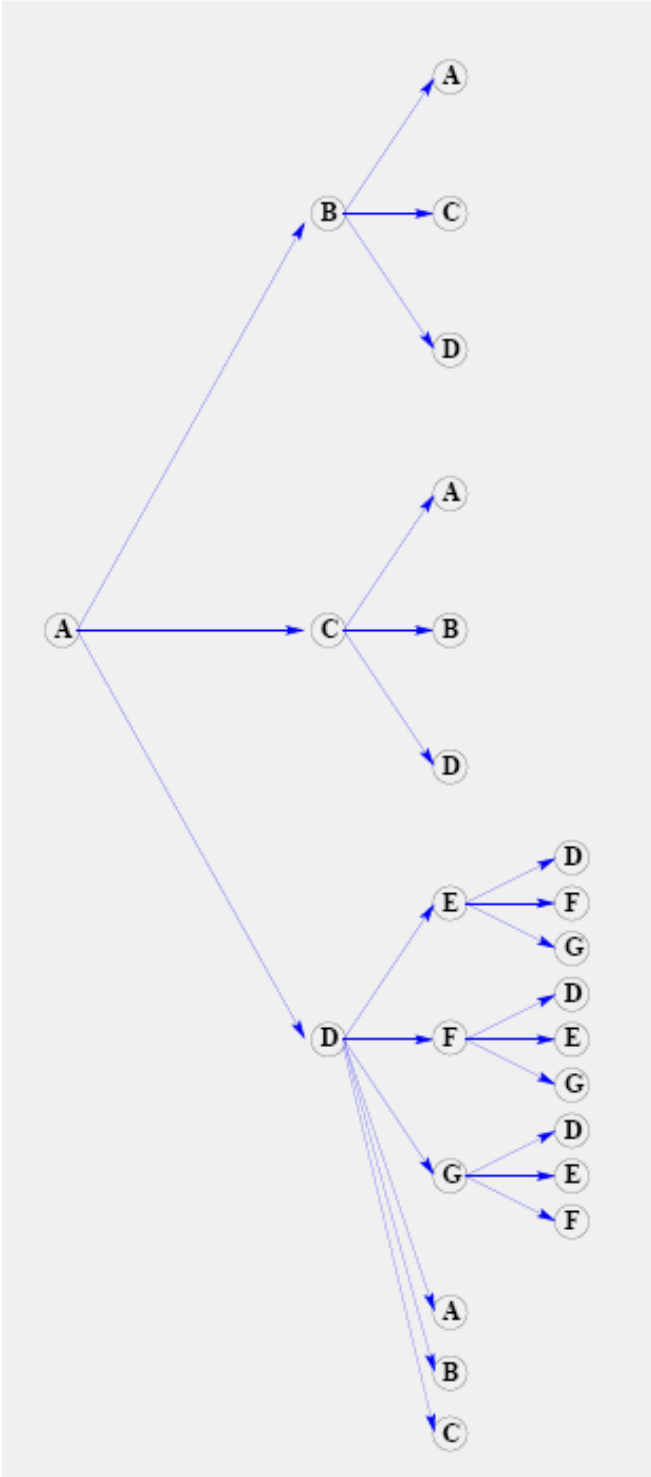


Figura 5.30 .*Grafo de mensajes que se inundan en la MANET de la figura anterior.*

6. Seguridad

Las redes ad hoc inalámbricas son naturalmente vulnerables a diversos tipos de ataques y los protocolos de encaminamiento ad hoc no son una excepción. Existen diversas propuestas para proveer de seguridad a los protocolos de encaminamiento actuales. Una de ellas la podremos ver en la sección 3.3.4 con la seguridad incorporada por el protocolo SAODV.

En el contexto de las redes ad hoc debemos distinguir entre diferentes tipos de nodos que intentarán vulnerar la seguridad de estas redes:

- Un nodo *malicioso* será un atacante que no puede autenticarse a sí mismo como nodo legítimo debido a la falta de información criptográfica válida.

- Un nodo *comprometido* será un atacante interno quien se comporta de forma malintencionada, sin embargo puede ser autenticado por la red como un nodo legítimo y obtener la confianza de los otros nodos.

- Un nodo *selfish* (egoísta/acaparador) será un nodo que tendrá tendencia a denegar la provisión de servicios en beneficio de otros nodos con el fin de conservar sus propios recursos.

A partir de aquí podemos hablar de las necesidades y de los atributos de seguridad requeridos por un protocolo seguro y del protocolo seguro en sí.

6.1. Vulnerabilidades

6.2. Requisitos de seguridad

Seguridad es un término habitual dentro de la terminología de redes de computadores. ¿Qué se entiende exactamente por seguridad o, de forma más concreta, que constituye la seguridad en redes ad hoc? Los requisitos básicos de seguridad, más o menos, son los mismos que las de redes cableadas. Lo que hay que estudiar es la aplicabilidad de las soluciones tradicionales en redes inalámbricas tradicionales y en la implementación de ciertos mecanismos en los que hay que tener especial cuidado. A continuación, introducimos brevemente los términos que se emplean cuando se discuten aspectos de seguridad en una red.

Disponibilidad. Disponibilidad significa que los servicios prestados por un nodo debe ser proporcionados siempre, incluso en caso de que se reciban ataques. Los nodos deben estar disponibles para la comunicación en todo momento. En otras palabras, se garantiza la supervivencia de los servicios de red en presencia de ataques de denegación de servicio, que pueden ser lanzadon desde cualquier capa de una red ad hoc, por ejemplo, a través del colapso del medio mediante emisiones de radio, o del agotamiento de las baterías.

Autenticidad. Autenticidad es la confirmación de que las partes implicadas en la comunicación son quienes dicen ser y no suplantadores. Esto requiere que los nodos dispongan de un sistema que les permita probar que son quienes dicen ser. Sin autenticación, un adversario podría suplantar sin ningún problema a otro, pudiendo acceder a información delicada o incluso clasificada. Además, este hecho probablemente interferirá en el normal y seguro funcionamiento de la red.

Confidencialidad. Un adversario no debe poder acceder a la información transmitida entre dos nodos. Esto garantiza que la información no es divulgada entre entidades no autorizadas. Para garantizar la confidencialidad, es necesario prevenir que nodos intermedios y nodos en los que se desconfía puedan procesar el contenido de los paquetes que son transmitidos. Si se emplean cuidadosamente los mecanismos de autenticación, proporcionar confidencialidad es un proceso sencillo.

Integridad. Integridad es la garantía de que un mensaje o paquete enviado no ha sido modificado durante el transporte o, dicho de otra forma, de que se recibe como ha sido enviado. Un mensaje puede ser corrompido por un nodo no malintencionado debido, por ejemplo, a algún problema en la propagación d ela señal de radio. Sin embargo, siempre existe la posibilidad, de que un nodo malicioso modifique intencionadamente el contenido de un mensaje. Es obligación de los mecanismos de seguridad detectar cuando ocurre esto.

No repudio. No repudio significa que el remitente de un mensaje no puede negar más adelante el envío de la información y que el receptor no puede negar la recepción. Esto puede ser útil mientras se detectan y aíslan nodos comprometidos. Cualquier nodo que reciba un mensaje erróneo puede acusar el remitente con la prueba y, así, convencer otros a nodos sobre el mal comportamiento del nodo comprometido. El modelo de confianza empleado por la red propaga actualizaciones desde nodos lejanos entre sí que posibilitan un mejor conocimiento del comportamiento de los nodos de la red.

Orden. Las actualizaciones recibidas de los routers deben estar en orden, la no concurrencia de éstas, puede afectar al correcto funcionamiento de los protocolos de enrutamiento. Los mensajes no deben reflejar el verdadero estado de la red ni propagar información falsa.

Puntualidad. Las actualizaciones de enrutamiento deben ser emitidas ordenadas de manera correcta. Los mensajes de actualización que lleguen más tarde pueden no mostrar correctamente el estado de los enlaces que conforman la red. Pueden causar envíos incorrectos o propagar información falsa que afecte a la credibilidad de la información de actualización. Si un nodo que transmite información entre dos grandes componentes se encuentra rodeado por vecinos maliciosos, parte de la red será inaccesible.

Aislamiento. El aislamiento requiere que el protocolo sea capaz de detectar los nodos malintencionados e impedir que interfieran en las comunicaciones de la red. El protocolo puede ser diseñado para ser inmune a nodos malintencionados.

Autorización. Un usuario o nodo autorizado dispone de una credencial proporcionada por la autoridad de certificación. Estas credenciales especifican los privilegios y permisos asociados a los usuarios o a los nodos. Sin embargo, estas credenciales no son

empleadas por los paquetes del protocolo de enrutamiento, y cualquier paquete puede provocar propagaciones de actualizaciones o modificar la tabla de enrutamiento.

Computación ligera. Muchos dispositivos conectados a una red ad hoc se alimentan mediante baterías y/o tienen una capacidad de computación limitada. Por lo tanto, no se puede esperar que un nodo realice cálculos muy complejos. Siendo necesario realizar operaciones como algoritmos criptográficos de clave pública o del camino más corto, deben implicar al menor número de nodos posible.

Privacidad de localización. A menudo la información almacenada en las cabeceras es procesable de la misma manera que el contenido del mensaje. El protocolo de enrutamiento debería proteger la información sobre la localización de los nodos y la estructura de la red.

Auto-estabilización. Un protocolo de enrutamiento debe ser capaz de recuperarse automáticamente de cualquier problema en un período finito de tiempo sin intervención humana. Esto supone que la red no se mantenga fuera de uso permanentemente tras la inyección de algunos paquetes corruptos. Si el protocolo es auto-estabilizado, un atacante que intente infligir un daño permanente en la red, deberá mantenerse conectado continuamente a la red, de forma que será fácilmente localizable.

Robustez bizantina. Un protocolo de enrutamiento debe funcionar correctamente aunque varios nodos que participen en el enrutamiento corrompan su funcionamiento. Es una versión más estricta de la auto-estabilización, el protocolo no sólo debe recuperarse automáticamente del ataque, si no que no debe dejar de funcionar durante el ataque. Obviamente, si un protocolo no posee la propiedad de auto-estabilización, no es presenta robustez bizantina.

Anonimato.Anonymity. Ni el nodo móvil ni el software del sistema deben exponer información que permita sacar alguna conclusión sobre el usuario actual o propietario del equipo. En el caso de usar identificadores de dispositivo o de red (dirección MAC, dirección IP), no es posible realizar una asociación entre el identificador y la identidad del propietario tanto para los compañeros de comunicaciones como para un atacante.

Administración de claves. Los servicios de administración de claves deben proporcionar respuestas a las siguientes preguntas:

- *Modelo de confianza:* cuantos elementos diferentes de la red pueden confiar unos en otros y confiar en las relaciones entre nodos de la red.
- *Criptosistemas:* Es conveniente utilizar criptosistemas de clave pública aunque sean sensiblemente más lentos que los criptosistemas de clave secreta con mismo nivel de seguridad.
- *Generación de claves:* que elementos pueden generar claves para sí mismos y para otros elementos de la red.

Control de acceso. El control de acceso consiste en gobernar la manera en la que los usuarios o de los usuarios virtuales como el proceso de un sistema operativo acceden a los datos. Sólo los nodos autorizados pueden formar, destruir, unirse o salir de grupos. El control de acceso puede consistir también en la manera en que los nodos se registran

en la red para poder comunicarse con otros nodos cuando acceden por primera vez a la red. Existen varios esquemas de control de acceso:

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role Based Access Control (RBAC)

Confianza. Si la seguridad física es baja y las relaciones de confianza son dinámicas, las posibilidades de que haya un fallo de seguridad crecen rápidamente**.**

6.3. Ataques

Se pueden distinguir dos niveles de ataque:

- Ataques sobre los mecanismos básicos de la red ad hoc, como el enrutamiento.
- Ataques sobre los mecanismos de seguridad y concretamente sobre el mecanismo de administración de claves.

Por otro lado, los ataques contra las redes ad hoc se pueden dividir en dos grupos:

- Ataques pasivos: normalmente solo incluyen la escucha de datos. Ejemplos de ataques pasivos son recubrimiento de canales, análisis del tráfico, sniffing de claves comprometidas, etc.
- Ataques activos: consisten en acciones realizadas por adversarios, por ejemplo, la replicación, modificación, y borrado de datos intercambiados. Los adversarios intentan activamente cambiar el comportamiento del protocolo con sus ataques. La información revelada a atacantes pasivos por los paquetes del protocolo puede ser utilizada para realizar ataques pasivos.

Los ataques externos son típicamente ataques activos que son perpetrados para, por ejemplo, causar congestión, propagar información de enrutamiento incorrecta, evitar que los servicios funcionen correctamente. Los ataques externos se pueden prevenir fácilmente usando mecanismos standard de seguridad como firewalls, encriptación y otros. Los ataques internos son normalmente ataques más severos ya que los nodos maliciosos forman parte de la red como partes autorizadas y por tanto están protegidos por los mecanismos de seguridad.

De los muchos ataques que pueden sufrir los protocolos de enrutamiento destacamos los siguientes:

Desbordamiento de la tabla de encaminamiento: tiene lugar cuando un nodo atacante anuncia rutas hacia nodos no existentes a los nodos autorizados en la red. El objetivo es causar un desbordamiento de las tablas de encaminamiento, que a su vez evitaría la creación de entradas que correspondan a rutas nuevas a los nodos autorizados. Los protocolos de encaminamiento proactivos son los más vulnerables a estos ataques comparados con los protocolos de encaminamiento reactivos.

Envenenamiento de tabla de encaminamiento: ocurre cuando los nodos comprometidos en las redes envían actualizaciones de rutas ficticias o modifican los paquetes de actualización de rutas genuinos enviados a otros nodos no comprometidos.

Este tipo de ataque puede hacer que el encaminamiento deje de ser óptimo, puede congestionar porciones de la red, o hasta hacerlas inaccesibles.

Replicación de paquetes: ocurre cuando un nodo atacante replica paquetes anteriores. Esto consume recursos de ancho de banda adicional y potencia de las baterías para los nodos y también causa un confusión innecesaria en el proceso de encaminamiento.

Envenenamiento de la caché de ruta: en el caso de protocolos de encaminamiento reactivos (como AODV), cada nodo mantiene una caché de rutas la que almacena información recolectada de las rutas se han hecho conocidas por el nodo en un pasado reciente. De la misma manera que en el envenenamiento de tabla de encaminamiento, un atacante puede también envenenar la caché de ruta para lograr objetivos similares.

Rushing Attack: los protocolos de encaminamiento reactivos que usan eliminación de duplicados durante el proceso de descubrimiento de ruta son vulnerables a estos ataques. Un nodo atacante que recibe un paquete Route REQuest (RREQ) a partir de un nodo fuente inunda con el paquete rápidamente toda la red antes de que los otros nodos que reciben el mismo paquete RREQ puedan reaccionar. Los nodos que reciben los paquetes RREQ legítimos asumen que éstos son duplicados del paquete ya recibido proveniente del nodo atacante y por lo tanto los descartan. Cualquier ruta descubierta por el nodo fuente podría contener al nodo atacante como uno de los nodos intermedios. Por lo tanto la fuente no podría ser capaz de encontrar rutas seguras, es decir, rutas que no contengan al nodo atacante.

Black Hole: ocurre cuando un nodo malicioso usa el protocolo de encaminamiento para anunciar que él posee la ruta más corta a los nodos de los paquetes que el quiere interceptar. En un protocolo basado en inundación o flooding, el atacante escucha las solicitudes de ruta, cuando recibe una solicitud para una ruta al nodo objetivo, el atacante crea una respuesta que consiste en una ruta extremadamente corta. Si la respuesta del nodo malicioso llega al nodo solicitante antes de que la respuesta del nodo actual, se crea una ruta falsa. Una vez que el nodo malicioso es capaz de insertarse entre la comunicación de los nodos, puede hacer cualquier cosa con los paquetes que se transmiten entre ellos, por ejemplo, elegir eliminar paquetes para realizar una denegación de servicios o usar su ubicación en la ruta como primer paso para un ataque main-in-the-middle.

Privación de sueño(sleep deprivation): este tipo de ataque es práctico sólo en aquella redes ad hoc donde la vida de las baterías es un parámetro crítico. Los dispositivos intentan conservar la energía de sus baterías sólo transmitiendo cuando es necesario. Un atacante puede intentar consumir baterías mediante la solicitud de rutas o mediante el reenvío innecesario de paquetes a un nodo, por ejemplo, mediante un ataque black hole.

Divulgación de ubicación: este ataque consiste en revelar algún tipo de información acerca de la localización de los nodos o la estructura de la red.

Otros ataques de tipo multicapa y que también atentan contra los protocolos de encaminamiento son:

Denegación de Servicios: un atacante procura evitar que los usuarios legítimos y autorizados de los servicios ofrecidos por la red tengan acceso a esos servicios. Este es

un tipo de ataque multicapa, pero en particular en la capa de red, un atacante podría tomar parte del proceso de encaminamiento y explotar el protocolo de encaminamiento para interrumpir el normal funcionamiento de la red.
Suplantación de identidad: un atacante asume la identidad y privilegios de un nodo autorizado, con el fin de utilizar recursos de la red que no están disponibles bajo circunstancias normales o para interrumpir su normal funcionamiento mediante la inyección de información de encaminamiento falsa. Un atacante podría enmascararse como un nodo autorizado usando varios métodos, por ejemplo el ataque de man-in-the-middle.

6.4. Mecanismos de seguridad

6.4.1. Administriación de claves

6.4.1.1. Infraestructura de clave pública (PKI)

El acrónimo PKI deriva de "Public Key Infrastructure" (Infraestructura de Clave Pública) y es la forma común de referirse a un sistema complejo necesario para la gestión de certificados digitales y aplicaciones de la Firma Digital.

Una PKI bien construida debe proporcionar:

- **Autenticidad**. La firma digital tendrá la misma validez que la manuscrita.
- **Confidencialidad**, de la información transmitida entre las partes.
- **Integridad**. Debe asegurarse la capacidad de detectar si un documento firmado ha sido manipulado.
- **No Repudio**, de un documento firmado digitalmente.

6.4.1.1.1. Criptosistemas

La criptografía es una ciencia cuyo objeto es transformar mediante convenciones secretas denominadas claves, informaciones o señales claras en informaciones o señales ininteligibles por terceros que no conozcan el secreto, o realizar la operación inversa gracias a medios, hardwares o softwares diseñados con este fin. La criptografía permite detectar la pérdida de integridad de informaciones, de autentificar interlocutores y proteger la confidencialidad de las informaciones.

La seguridad de la criptografía se basa en tres factores:

- La calidad de los algoritmos utilizados. No es indispensable que estos algoritmos sean confidenciales, pero deben apoyarse en preguntas consideradas difíciles de resolver por los matemáticos, si no conocen un secreto. Este secreto se denomina clave.

- La implementación de estos algoritmos. Es mucho más fácil evitar una mala implementación de un algoritmo que "romper" el propio algoritmo.

- La gestión de las claves. Cuando se utiliza un mismo secreto entre dos usuarios (sistema simétrico), la multiplicación del número de usuarios aumenta el número de secretos que compartir de manera cuadrática. En cambio, la utilización de un sistema donde no se necesita compartir un secreto (sistema asimétrico) facilita la gestión de los secretos pero no evita la protección de la integridad de las claves ni la exigencia de protección de la confidencialidad de los secretos propios de los usuarios.

Un Criptosistema se define como la quíntupla (m,C,K,E,D), donde:

- m representa el conjunto de todos los mensajes sin cifrar (texto plano) que pueden ser enviados.
- C Representa el conjunto de todos los posibles mensajes cifrados, o criptogramas.
- K representa el conjunto de claves que se pueden emplear en el Criptosistema.
- E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de m para obtener un elemento de C. Existe una transformación diferente E_k para cada valor posible de la clave K.
- D es el conjunto de transformaciones de descifrado, análogo a E.

Todo Criptosistema cumple la condición $D_k(E_k(m))=m$ es decir, que si se tiene un mensaje m, se cifra empleando la clave K y luego se descifra empleando la misma clave, se obtiene el mensaje original m.

Existen dos tipos fundamentales de Criptosistemas utilizados para cifrar datos e información digital y ser enviados posteriormente después por medios de transmisión libre.

- **Simétricos o de clave privada:** se emplea la misma clave K para cifrar y descifrar, por lo tanto el emisor y el receptor deben poseer la clave. El mayor inconveniente que presentan es que se debe contar con un canal seguro para la transmisión de dicha clave.
- **Asimétricos o de llave pública:** se emplea una doble clave conocidas como K_p (clave privada) y K_P (clave Pública). Una de ellas es utilizada para la transformación E de cifrado y la otra para el descifrado D. En muchos de los sistemas existentes estas clave son intercambiables, es decir que si empleamos una para cifrar se utiliza la otra para descifrar y viceversa.

Los sistemas asimétricos deben cumplir con la condición que la clave Pública (al ser conocida y sólo utilizada para cifrar) no debe permitir calcular la privada. Como puede observarse este sistema permite intercambiar claves en un canal inseguro de transmisión ya que lo único que se envía es la clave pública.

Los algoritmos asimétricos emplean claves de longitud mayor a los simétricos. Así, por ejemplo, suele considerarse segura una clave de 128 bits para estos últimos pero se recomienda claves de 1024 bits (como mínimo) para los algoritmos asimétricos.

Esto permite que los algoritmos simétricos sean considerablemente más rápidos que los asimétricos.

En la práctica actualmente se emplea una combinación de ambos sistemas ya que los asimétricos son computacionalmente más costosos (mayor tiempo de cifrado). Para realizar dicha combinación se cifra el mensaje m con un sistema simétrico y luego se encripta la clave K utilizada en el algoritmo simétrico (generalmente más corta que el mensaje) con un sistema asimétrico.

Después de estos Criptosistemas modernos podemos encontrar otros no menos importantes utilizados desde siempre para cifrar mensajes de menos importancia o domésticos, y que han ido perdiendo su eficacia por ser fácilmente criptoanalizables y por tanto "reventables". Cada uno de los algoritmos clásicos descriptos a continuación utilizan la misma clave K para cifrar y descifrar el mensaje.

6.4.1.1.2. Criptosistemas simétricos

La mayoría de los algoritmos simétricos actuales se apoyan en los conceptos de Confusión y Difusión vertidos por Claude Shannon sobre la Teoría de la Información a finales de los años cuarenta.

Estos métodos consisten en ocultar la relación entre el texto plano, el texto cifrado y la clave (Confusión); y repartir la influencia de cada bit del mensaje original lo más posible entre el mensaje cifrado (Difusión).

El objetivo del presente no es entrar en detalles de cada uno de los muchos algoritmos existentes, por lo que sólo se dará una idea de su funcionamiento y complejidad.

Redes de Feistel

Este algoritmo no es un algoritmo de cifrado per se, pero muchos de los vistos a continuación lo utilizan como parte vital en su funcionamiento. Se basa en dividir un bloque de longitud n (generalmente el texto a cifrar) en dos mitades, L y R. Luego se define un cifrado de producto iteractivo en el que la salida de cada ronda es la entrada de la siguiente.

DES

Data Encryption Standard es el algoritmo simétrico más extendido mundialmente. A mediados de los setenta fue adoptado como estándar para las comunicaciones seguras (Estándar AES) del gobierno de EE.UU. En su principio fue diseñado por la NSA (National Security Agency) (1) para ser implementado en hardware, pero al extenderse su algoritmo se comenzó a implementar en software.

DES utiliza bloques de 64 bits, los cuales codifica empleando claves de 56 bits y aplicando permutaciones a nivel de bit en diferentes momentos (mediante tablas de permutaciones y operaciones XOR). Es una red de Feistel de 16 rondas, más dos permutaciones, una que se aplica al principio y otra al final.

La flexibilidad de DES reside en que el mismo algoritmo puede ser utilizado tanto para cifrar como para descifrar, simplemente invirtiendo el orden de las 16 subclaves obtenidas a partir de la clave de cifrado.

En la actualidad no se ha podido romper el sistema DES criptoanalíticamente (deducir la clave simétrica a partir de la información interceptada). Sin embargo una empresa española sin fines de lucro llamado Electronic Frontier Foundation (EFF) (2) construyo en Enero de 1999 una máquina capaz de probar las 2 56 claves posibles en DES y romperlo sólo en tres días con fuerza bruta.

A pesar de su caída DES sigue siendo utilizado por su amplia extensión de las implementaciones vía hardware existentes (en cajeros automáticos y señales de video por ejemplo) y se evita tener que confiar en nuevas tecnologías no probadas. En vez de abandonar su utilización se prefiere suplantar a DES con lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave.

6.4.1.1.3. Criptosistemas asimétricos

Los Criptosistemas de clave pública fueron inventados a finales de los años 70, con ayuda del desarrollo de la teoría de complejidad alrededor de esa época.

Observando que basados en la dificultad de un problema y de los miles de años que llevaría resolverlo y con un poco de suerte, se observó que un criptosistema podría ser desarrollado teniendo dos claves, una privada y una pública.

Con la clave pública se puede cifrar mensajes, y descifrarlos con la clave privada. Así el propietario de la clave privada sería el único que podría descifrar los mensajes, pero cualquier persona que conozca la clave pública podría enviarlos en forma privada.

Otra idea que se observó fue la del intercambio de claves. En una comunicación entre dos partes sería de mucha utilidad generar una clave secreta común para cifrado a granel usando un criptosistema de clave secreta (por ejemplo, cifrado de bloques).

De hecho, Whitfield Diffie y Martin Hellman utilizaron ideas de la teoría de números para construir un protocolo de intercambio de claves, el cual dio comienzo a la era de los criptosistemas de clave pública.

Poco después, Ronald Rivest, Adi Shamir, y Leonard Adleman desarrollaron un criptosistema que fue el primer criptosistema de clave pública real, capaz de cifrar y manejar firmas digitales.

Más adelante fueron apareciendo otros criptosistemas de clave pública que usaron diferentes ideas subyacentes (por ejemplo, los problemas de la mochila, diversos grupos en campos finitos, y los enrejados).

Muchos de ellos resultaron ser inseguros. Sin embargo, el protocolo Diffie-Hellman y el RSA parecen ser los dos más fuertes hasta el momento.

La dificultad computacional del problema es el ingrediente básico en cualquier criptosistema de clave pública. La seguridad de los criptosistemas esta basada en el hecho de que la clave privada puede ser computada desde la clave pública únicamente solucionando este difícil problema. A continuación presentaremos alguna terminología relevante usada en la criptografía de clave pública.

A continuación, vamos a analizar los algoritmos cifradores de clave pública que más repercusión han tenido en los últimos años, así como algunos ejemplos de sistemas que están siendo objeto de estudio en la actualidad. Presentamos estos ejemplos clasificados por el fundamento matemático en el que basan su funcionamiento.

2.4.4.1.1.1. Factorización

2.4.4.1.1.1.1. RSA

El algoritmo cifrador RSA toma su nombre de las iniciales de los apellidos de sus creadores. En 1977 Ron Rivest, Adi Shamir y Len Adleman describen en el MIT un sistema criptográfico con clave pública, cuya existencia no se revelaría hasta 1997 ya que era confidencial. Anteriormente, Clifford Cocks, un matemático británico rabajando para la agencia de inteligencia británica GCHQ describió un sistema equivalente en un documento interno en 1973. Se trata de un algoritmo asimétrico cifrador por bloques, que emplea una clave pública es distribuida entre los individuos de los que se desea recibir información, y otra privada, que se mantiene en secreto, siendo sólo conocida por su propietario.

El funcionamiento del criptosistema RSA se basa en la dificultad para factorizar grandes números enteros. Las claves RSA son más complejas, y brindan mayor seguridad, cuando mayor es su tamaño. Originalmente se empleaban claves de 160(RSA 160) o 200 (RSA200) bits de largo. Pero la velocidad con la que avanzaba la informática provocó que rápidamente se pasase a RSA 576, RSA 1024, RSA 2048, etc. Hoy día existe software capaz de generar RSA8192.

Generación de claves

El algoritmo de generación de claves es el siguiente:

Una vez elegida la longitud l de clave, se seleccionan dos números primos largos p y q de manera que $p \neq q$ y su producto $n = pq$ tiene longitud l.

Calcular $\phi(n) = (p - 1)(q - 1)$, donde Φ es la función φ de Euler.

Seleccionar un entero positivo e tal que $1 < e < \phi(n)$ tales que e y $\varphi(n)$ sean primos entre sí.

Calcular d tal que $de \equiv 1 \pmod{\phi(n)}$.

Los números primos pueden ser comprobados con un test de primalidad (test probabilístico).

2.4.4.1.1.2. Logaritmos discretos

Se conoce como logaritmo discreto de *x* en base *a* módulo *n* a resolver la ecuación $x = a^y \bmod n$ donde *x*,*n* y *a* son constantes e *y* es la incógnita. El cálculo de $x = a^y \bmod n$ (donde n es un número primo de gran longitud) tiene un buen número de aplicaciones criptográficas en el campo de la criptografía asimétrica o de clave pública.

La utilidad de $a^y \bmod n$ es que su cálculo es bastante sencillo y eficiente, pero su inversa, es decir, obtener el valor y tal que $a^y \bmod n$ sea igual a una cantidad determinada, constituye un problema que exige grandes recursos computacionales para su resolución. Analizamos a continuación los principales ejemplos de algoritmos criptográficos de clave pública basados en algoritmos discretos.

Diffie – Hellman. Es un protocolo normalmente utilizado para el intercambio de llaves. En muchos protocolos criptográficos dos partes desean establecer comunicación. Sin embargo, asumen que inicialmente no poseen un secreto y de esta manera no pueden utilizar un criptosistema de llave secreta. El intercambio de llaves por el protocolo Diffie-Hellman remedia esta situación permitiendo la constricción de una llave secreta común sobre un canal de comunicaciín inseguro.

Se basa en un problema relacionado con logaritmos naturales, llamado el problema Diffie-Hellman. Este problema es considerado difícil, y es en algunas instancias tan difícil como el problema de logaritmos discretos.

El protocolo Diffie-Hellman se considera generalmente seguro cuando se utilizan grupos matemáticos apropiados. En particular, el elemento generador utilizado en la "exponenciación" debería tener un período grande.

Algoritmos de logaritmos discretos se pueden utilizar para realizar ataques contra Diffie-Hellman, y con ataques pasivos que es lo mejor de lo actualmente posible, asumiendo la elección de parámetros correctos.

Se pueden generar problemas sutiles por la mala elección del generador. Se han escrito muchos artículos sobre los problemas que pueden surgir. Una de las referencias más notorias es Oorschot and Wiener's en Acuerdos de llave Diffie-Hellman con exponentes pequeños (Eurocrypt '96).

Ataques contra Diffie-Hellman también incluyen el ataque man in the middle. Este ataque requiere intervención adaptativa, pero es muy fácil en la práctica si el protocolo no utiliza contramedidas tales como firmas digitales.

Diffie-Hellman usualmente no es implementado en hardware, y así los ataques de hardware no son una amenaza importante. Este puede cambiar en el futuro, cuando las comunicaciones móviles se extiendan más.

Cuando dos terminales desean comunicarse a través de un canal seguro mediante el algoritmo Diffie-Hellman, realizan el siguiente proceso.

1. Ambos terminales seleccionan un número primo grande p y un número g, con $1 < g < p$,

2. El terminal 1 selecciona en secreto un entero n.

3. El terminal 2 selecciona en secreto un entero m.

4. El terminal 1 envía al terminal 2 $ng \pmod p$, el producto de ng reducido a módulo p.

5. El terminal 2 envía al terminal 1 $mg \pmod p$.

6. La clave secreta es $s = nmg \pmod p$ que puede ser calculada fácilmente por ambos terminales.

Ambos terminales pueden comunicarse de forma segura porque ambos pueden calcular fácilmente s:

- Ambos conocen p, g, $ng \pmod p$, y $mg \pmod p$.
- Usando el algoritmo Euclídeo se encuentra de forma rápida a, $b \in Z$ tal que $ag + bp = 1$, el cual existe porque $\text{mcd}(g, p) = 1$ (Debido a que g y p son coprimos).
- A continuación, $ang \equiv n \pmod p$ se puede calcular fácilmente. Por lo tanto, ambos pueden encontrar la clave secreta de una manera eficiente.

Vamos a mostrar un ejemplo del algoritmo con números pequeños (aunque en la práctica se utilizan números de 200 dígitos).

1. $p = 97$, $g = 5$.

2. $n = 31$.

3. $m = 95$.

4. $ng = 58 \pmod{97}$.

5. $mg = 87 \pmod{97}$.

6. $s = nmg = 78 \pmod{97}$.

Sin embargo, la comunicación es segura en caso de que los posibles “espías” desconozcan el algoritmo. En caso de conocerlo, al poder escuchar los valores p, g, ng y mg, podrán calcular de manera sencilla la clave secreta s. De esta forma, podrán descifrar sin ningún problema los mensajes intercambiados.

Por lo tanto, fue necesario revisar el algoritmo en búsqueda de una solución segura. Finalmente, propusieron el algoritmo que describimos a continuación.

1. Ambos dispositivos eligen un número primo p de 200 dígitos y un número g con $1 < g < p$.

2. El terminal 1 elige un número secreto n.

3. El terminal 2 elige un número secreto m.

4. El terminal 1 calcula $g^n \pmod p$ y lo transmite al terminal 2.

5. El terminal 2 transmite $g^m \pmod p$ al terminal 1.

6. La clave secreta es:

$$s \equiv (g^n)^m \equiv (g^m)^n \equiv g^{nm} \pmod p.$$

DSS (Digital Signature Standard - Estandar de Firma Digital). Una mecanismo de unicamente-firma avalado por el gobierno de los Estados Unidos. El algoritmo DSA (Digital Signature Algorithm - Algoritmo de Firma Digital) fundamental es similar al algoritmo de firma utilizado por ElGamal o por el Schnorr. También es bastante eficiente, pero no tanto como el RSA para verificación de firma. El estandar define el DSS para utilizar la función hash SHA-1 exclusivamente para calcular el resúmen del mensaje.

El problema principal con el DSS es el tamaño del subgrupo fijo (el orden del elemento generador), el cual limita la seguridad a alrededor de sólo 80 bits. Los ataques de hadware pueden tener que ver con algunas implementaciones de DSS. Sin embargo, es ampliamente aceptado y utilizado como un buen algoritmo.

El codificador de llave publica ElGamal: este es una extensión de la idea original de Diffie/Hellman de compartir la generación secreta. Escencialmente, genera un secreto compartido y lo utiliza como un relleno por única vez para encriptar un bloque de dato. ElGamal es el predecesor del DSS y hoy es perfectamente utilizable, a pesar de que no se han creado amplios estandares para él.

Criptosistemas de Curva Elíptica: son otra forma de implementar los métodos de logaritmos discretos. Una curva elíptica es basicamente un conjunto de puntos que satisfacen la ecuación y^2=x^3 + ax + b cuando se los considera dentro de un campo finito de características p (donde p debe ser mayor que 3). Una ecuación ligeramente diferentese necesita para el caso de características pequeñas, p=2 y p=3.

Los puntos en una curva elíptica pueden ser agregadas conjuntamente y forman una estructura denominada grupo (en realidad un grupo abeliano). Es justo una manera de decir que puedes hacer aritmética con ellos como lo puedes hacer con enteros cuando solo utilizas adiciones y sustracciones.

En consideración a la criptografía, las curvas elípticas tienen varios beneficios teóricos y también prácticos.

Un beneficio práctico de la no existencia de un algoritmo discreto de cálculo para curvas elípticas es que el tamaño de la llave, así como también la firma digital producida y el mensaje encriptado son menores. Efectivamente, una manera simple de calcular el límite se seguridad para el tamaño de la llave es tomar un tamaño de la llave para el criptosistema de llave secreta en bits y luego solo multiplicarlo por 2. Esto da una estimación gruesa, esto es bueno al momento de una instancia genérica de curvas elípticas.

Las curvas elípticas pueden ser implemntadas eficientemente en hardware y software, y compiten bien en velocidad con criptosistemas tales como el RSA y el DSS. Existen varios intentos de estandarización para criptosistemas de curvas elípticas (por ejemplo, ECDSA y ANSI). Por el momento las curvas elípticas son muy conocidas, pero no muy utilizadas en la práctica.

La seguridad de los criptosistemas de curvas elípticas ha permanecido estable por años, a pesar de que se han logrado muchos avances significativos en ataques contra instancias especiales. No obstante, esto ha sido conjeturado por los investigadores hace ya varios años y no han surgido grandes sorpresas todavía.

El algoritmo XTR presentado recientemente por Lenstra and Verheul podría convertirse en un buen competidor para las curvas elípticas. Sin embargo, las curvas elípticas parecen ser levemente mejores en performance y definitivamente trata mejor el tamaño de la llave.

LUC es un criptosistema de llave pública que utiliza un grupo especial basado en la secuencia de LUCAS (relacionado a las Sucesiones de Fibonacci)como su bloque de construcción básico. Puede implementar todos los algoritmos basados en logaritmos discretos, y en un sentido LUC es una clase de algoritmo de llave pública.

Los diferentes algoritmos de llave pública basados en la aritmética de LUC son llamados LUCDIF (LUC Diffie-Hellman), LUCELG (LUC ElGamal), y LUCDSA (LUC Digital Signature Algorithm). Varios de estos estan patentados.

Sin embargo, actualmente parece haber pocas razones para utilziar los criptosistemas LUC, ya que ofrecen pocos beneficios sobre las curvas elípticas o XTR.

XTR es un criptosistema de llave pública desarrollada por Arjen Lenstra y Eric Verheul. XTR es similar a LUC ya que utiliza un grupo multiplicativo específico de un campo finito particular (de hecho Fp^6) como su grupo.

Sin embargo, XTR tiene características de novela tales como necesitar aproximadamente solo 1/3 de los bits para las firmas y mensajes encriptados. Esto es obtenido utilizando una representación específica rastreada de los elementos de ese grupo, y realizando todos los cálculos utilizando esta representación.

Todos los algorítmos de llave pública basados en logaritmos discretos pueden ser implementados con las ideas del XTR. Así de una manera XTR es un nombre genérico para una clase de algoritmo de llave pública, similar al LUC.

Tal vez sorprendentemente, el algoritmo es eficiente y de acuerdo a Lenstra y Verheul puede ser un buen substituto de las curvas elípticas, DSS y aún del RSA. Tiene la ventaja sobre las curvas elípticas de que está basado escencialmente en el mismo problema de logaritmo discreto como el DSS, el cual puede ayudar a los criptógrafos y otros a aceptar que es rápido como algoritmos fuerte.

2.4.4.1.1.3. Mochilas

Existen pocos de criptosistemas de Mochila de llave pública interesantes, ninguno de los cuales es de particular importancia.

- El **criptosistema Rivest-Chor** está basado en una variente particular del problema de mochilas. Este es uno de los criptosistemas de mochilas que tiene mejor resistencia a los ataques.
- **Merkle-Hellman:** Este fue el primer criptosistema de mochilas. Se basaba en la simple idea de ocultar el fácil y super-incremental problema de mochilas con máscaras. Sin embargo, fue roto en 1980.

2.4.4.1.1.4. Enrejado

En los últimos años el interes se ha dirigido hacia criptosistemas basados en enrejados. Una de las razones es que ciertas clases de problemas de enrejados son duros, y se han propuesto muchos criptosistemas eficientes y parecen ser fuertes.

- **NTRU** es un sistema critográfico propuesto a mediados de 1990 como un cifrador de llave pública eficiente. Se basa en el problema de enrejado, y tiene algunas características interesantes

Algunas de las versiones iniciales tenían problemas, pero la versión actual ha sido propuesta para algunos estandares de los Estados Unidos.

6.4.1.1.4. Firma digital

Una Firma Digital tiene dos características principales:

- Sólo puede ser generada por el poseedor de la clave privada y puede ser verificada por cualquiera que conozca la clave pública del firmante.
- Es dependiente del documento a firmar (la Firma Digital de un documento no puede emplearse para firmar otro documento).

El proceso de generación de una Firma Digital consiste en dos pasos:

- Empleando un algoritmo de "Hashing" se genera un resumen, de tamaño fijo, del documento.
- Se cifra el Hash empleando la clave privada del usuario.

Deberá transmitirse o almacenarse el documento original y la firma.

Deben realizarse los siguientes pasos:

- A partir del Documento Original, se genera de nuevo el Hash.
- Empleando la Clave Pública del firmante, se descifra la firma digital.
- Se comprueba si ambos "Hashes" coinciden, si es así, la firma es auténtica, si no lo es, el documento ha sido modificado y/o la firma es falsa.
- Adicionalmente, debe comprobarse que el Certificado Digital es válido. El Certificado puede haber caducado, o puede haber sido revocado por una de las partes (ver más adelante).

6.4.1.1.5. Certificados digitales

Las técnicas anteriormente indicadas, si bien son técnicamente correctas, implican un grave problema a nivel de seguridad: ¿Cómo se puede asegurar que una clave pública pertenece a un usuario dado?. Es necesario poder vincular la clave pública de un usuario con su identidad y para esto surge el concepto de "Certificado Digital", que contiene la siguiente información:

- Identidad del usuario (Nombre, NIF, etc...).
- Clave Pública del usuario.
- Periodo de Validez del Certificado.
- Identidad de la Autoridad Certificadora (entidad que emite el certificado).
- Firma digital del certificado (los datos anteriores más otras posibles extensiones personalizables, p.e. la dirección de correo electrónico), generada por la Autoridad Certificadora.

Esta información se encapsula en un formato estándar, definido por la norma ISO X.509 versión 3. Generalmente existirá un repositorio (p.e. directorio LDAP) en el que se publican todos los certificados gestionados por la PKI y puede ser consultado por otros usuarios de la PKI que quieran enviar información cifrada o verificar firmas digitales.

6.4.1.1.6. Autoridad certificadora

La Autoridad Certificadora, es la entidad que asegura la identidad de los usuarios de los certificados digitales. Posee su propio par de claves y firma digitalmente los certificados con su clave privada. Confiando en la Firma Digital de la Autoridad Certificadora, puede confiarse en cualquier certificado generado por la misma.

Las tareas realizadas por la Autoridad Certificadora son, entre otras, las siguientes:

- Procesa peticiones de Certificado a través de la Autoridad de Registro. Estas solicitudes están compuestas básicamente por los datos identificativos y la clave pública del solicitante.
- Genera los Certificados y los almacena en el repositorio público (p.e. LDAP).
- Gestiona la caducidad y renovación de certificados.
- Gestiona la revocación de certificados (p.e. por compromiso de la clave privada del usuario al serle sustraída su SmartCard).

Toda la fiabilidad de la Autoridad de Certificación se basa en la inviolabilidad de su propia clave privada, la cual resulta crítico proteger empleando medios técnicos y humanos.

6.4.1.1.7. Autoridad de registro

En toda PKI deben establecerse los mecanismos para que los usuarios soliciten su propio certificado, de tal forma que se asegure la identidad de dicho usuario. A este procedimiento se le denomina "Proceso de Registro" y se realiza a través de la denominada "Autoridad de Registro".

Existen dos tipos principales de registro:

- Registro Clásico. El solicitante acude en persona a una "Oficina de Registro", donde, tras acreditar su identidad, se le proporciona de forma segura su clave privada y su certificado.
- Registro Remoto. El usuario, a través de Internet, realiza una solicitud de certificado. Para esto empleará un software (p.e. un navegador) que generará el par de claves y enviará su clave pública a la Autoridad de Registro para que sea firmada por la Autoridad Certificadora y le sea devuelto su certificado.

La validez de la Firma Digital estará condicionada por la calidad del proceso de registro, siendo obligatorio para asegurar la validez legal de la firma, algún tipo de registro "Cara a Cara", ya que es el único que asegura la identidad del solicitante. Por otra parte, la validez de la firma digital también estará condicionada a la firma manuscrita de un "contrato" por el que el solicitante acepta su certificado y las condiciones de uso del mismo.

La Autoridad de Registro se compondrá de una serie de elementos tecnológicos (hardware y software específico) y unos medios humanos (los Operadores de Registro). Es el punto de comunicación entre los usuarios de la PKI y la Autoridad certificadora.

6.4.2. Autenticación

Autenticación o autentificación, en términos de seguridad de redes de datos, se puede considerar uno de los tres pasos fundamentales (AAA). Cada uno de ellos es, de forma ordenada:

- **Autenticación** En la seguridad de ordenador, la autenticación es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse. El remitente siendo autenticado puede ser una persona que usa un ordenador, un ordenador por sí mismo o un programa del ordenador. En un web de confianza, "autenticación" es un modo de asegurar que los usuarios son quién ellos dicen que ellos son - que el usuario que intenta realizar funciones en un sistema es de hecho el usuario que tiene la autorización para hacer así.
- **Autorización** Proceso por el cual la red de datos autoriza al usuario identificado a acceder a determinados recursos de la misma.

- **Auditoría** Mediante la cual la red o sistemas asociados registran todos y cada uno de los accesos a los recursos que realiza el usuario autorizados o no.

El problema de la autorización a menudo, es idéntico a la de autenticación; muchos protocolos de seguridad extensamente adoptados estándar, regulaciones obligatorias, y hasta estatutos están basados en esta asunción. Sin embargo, el uso más exacto describe la autenticación como el proceso de verificar la identidad de una persona, mientras la autorización es el proceso de verificación que una persona conocida tiene la autoridad para realizar una cierta operación. La autenticación, por lo tanto, debe preceder la autorización. Para distinguir la autenticación de la autorización de término estrechamente relacionada, exiten unas notaciones de taquigrafía que son: A1 para la autenticación y A2 para la autorización que de vez en cuando son usadas,también exiten los términos AuthN y AuthZ que son usados en algunas comunidades.

Los métodos de autenticación están en función de lo que utilizan para la verificación y estos se dividen en tres categorías:

- Sistemas basados en algo conocido. Ejemplo, un *password* (Unix) o *passphrase* (PGP).
- Sistemas basados en algo poseído. Ejemplo, una tarjeta de identidad, una tarjeta inteligente(*smartcard*), dispositivo usb tipo epass token, smartcard o dongle criptográfico.
- Sistemas basados en una característica física del usuario o un acto involuntario del mismo: Ejemplo, verificación de voz, de escritura, de huellas, de patrones oculares.

Cualquier sistema de identificación ha de poseer unas determinadas características para ser viable:

- Ha de ser fiable con una probabilidad muy elevada (podemos hablar de tasas de fallo de en los sistemas menos seguros).
- Económicamente factible para la organización (si su precio es superior al valor de lo que se intenta proteger, tenemos un sistema incorrecto).
- Soportar con éxito cierto tipo de ataques.
- Ser aceptable para los usuarios, que serán al fin y al cabo quienes lo utilicen.

La mayor parte de los sistemas informáticos y redes mantienen de uno u otro modo una relación de identidades personales (usuarios) asociadas normalmente con un perfil de seguridad, roles y permisos. La autenticación de usuarios permite a estos sistemas asumir con una seguridad razonable que quien se está conectando es quien dice ser para que luego las acciones que se ejecuten en el sistema puedan ser referidas luego a esa identidad y aplicar los mecanismos de autorización y/o auditoría oportunos.

El primer elemento necesario (y suficiente estrictamente hablando) por tanto para la autenticación es la existencia de identidades biunívocamente identificadas con

un identificador único (valga la redundancia). Los identificadores de usuarios pueden tener muchas formas siendo la más común una sucesión de caracteres conocida comúnmente como login.

El proceso general de autenticación consta de los siguientes pasos:

- El usuario solicita acceso a un sistema.
- El sistema solicita al usuario que se autentique.
- El usuario aporta las credenciales que le identifican y permiten verificar la autenticidad de la identificación.
- El sistema valida según sus reglas si las credenciales aportadas son suficientes para dar acceso al usuario o no.

6.4.3. Detección de intrusos

Un intruso, en una red Ad-Hoc, consiste en un nodo (dispositivo móvil, medio de transporte, satélite, etc.) que irrumpe en una red privada, a la que no pertenece, y accede a los datos y a la información que circula en dicha red. Su ingreso ilícito a la red, no es garantía de una fácil detección, teniendo en cuenta que se adquiere un acceso inalámbrico que constantemente cambia de topología. Además, una posible detección no es indemnidad para la red, pues es necesario después de detectar al intruso, eliminar su comunicación dentro de la red y excluirlo por completo de la red para evitar una nueva intrusión en futuras ocasiones.

En cuanto a las técnicas de Detección de Intrusos podemos nombrar, algunos modelos computacionales que ya hacen parte de este novedoso tema como lo es la seguridad en las redes Ad-Hoc. Existen propuestas de arquitecturas distribuidas y cooperativas para la detección de intrusos usando modelos de detección de anomalías y comportamientos incongruentes. Anomalías tales como incoherencias en la tabla de enrutamiento de un nodo, el flujo de desvío de información por causa de mensajes de difusión que capturan las direcciones físicas MAC de los nodos pertenecientes a la red por parte del nodo intruso, y la disrupción, se convierten en características que hacen pensar a un IDS de la red, que un nodo con tales comportamientos sea un nodo intruso.
Es por eso que en [32] se habla de un sistema propuesto, donde cada uno de los nodos de la MANET ejecuta un agente simple IDS que supervisa de forma constante las actividades locales al nodo. De acuerdo a las características de las anomalías se puede llegar a pensar en la existencia de un intruso, (Tarea que realiza el agente IDS del nodo correspondiente). Si el IDS detecta una intrusión a partir de los parámetros establecidos locales para el nodo, se ejecuta una operación de contradicción al intruso. En medio de este proceso puede suceder que a pesar de la anomalía detectada, el IDS no tenga la certeza definitiva de que exista propiamente una intrusión, de esta manera se inicia un proceso conjunto con los enrutadores o nodos que se encuentren más cercanos al nodo que detecto la anomalía, pero que no tiene la certeza de la intrusión; así con esta operación conjunta se pretende decidir si finalmente existe o no un ataque. Cuando se llega a la detección definitiva en la red se elimina la intrusión reiniciando la comunicación entre los nodos que si hacen parte de la red.

En este enfoque se pretende también incentivar a una búsqueda más avanzada de la intrusión, dando una estructura llamada "estructura multicapa", para realizar la detección del ataque, en las capas de los protocolos que hacen parte de la red.

Es precisamente en este tipo de enfoques de agentes sencillos aplicados a cada uno de los nodos para detección de anomalías y por consiguiente, realizar un rastreo a una posible intrusión, aparece un nuevo enfoque inspirado del anterior llamado: "Detección de Intrusos usando Agentes Móviles". En este modelo computacional, muy pertinente para topologías dinámicas, lo que se pretende es implementar un Sistema de Detección de Intrusos cimentado en técnicas computacionales con agentes móviles que se adaptan al medio, aún cuando este varía su estructura con el tiempo, y realizar una percepción de acuerdo a las posibles anomalías que presentan la presencia de un intruso. Básicamente podemos hablar de un agente móvil como una entidad software, muy sencilla de implementar, con comportamiento inteligente y que se adapta a su entorno dinámicamente viajando por la red MANET y ejecutándose sobre ciertos nodos.

Esto se encuentra con más detalle en [33] donde se propone una arquitectura distribuida y colaborativa en donde se asignan funciones de detección a diferentes tipos de agentes. Así, de esta forma se obtiene una carga de funcionalidades entre agentes viajando por los diferentes nodos.

Estas son las técnicas propuestas hasta el momento que reúnen las opciones útiles para realizar detección de intrusos en redes Ad-Hoc. Al final, siempre tendría que pensarse en implementar este tipo de modelos de detección de intrusos tendiendo en cuenta la ubicación de los nodos, y la necesidad que cobija esa red, esto sin olvidar que los dispositivos involucrados dispongan de la suficiente capacidad y autonomía como para que la ejecución de un sistema de detección no atribuya una limitación no resistible sobre las prestaciones ofrecidas al usuario final.

6.4.4. Enrutamiento seguro

La operación segura de un protocolo de encaminamiento de una red MANET es de principal importancia debido a la ausencia de una infraestructura fija. En su lugar, los nodos se asocian transitoriamente y cooperan virtualmente con cualquier otro nodo, incluyendo aquellos que podrían potencialmente interrumpir las operaciones de descubrimiento de ruta y reenvío de datos. En particular, la interrupción del descubrimiento de ruta puede ser un medio efectivo para obstaculizar sistemáticamente el flujo de datos. Los atacantes pueden responder con respuestas de rutas obsoletas o corruptas o difundir paquetes de control falsos para obstruir la propagación de consultas y actualizaciones de rutas legítimas.

Una vez vistos las propiedades de seguridad de una red ad hoc y los ataques que puede sufrir, particularmente contra la información de encaminamiento, los requisitos fundamentales que deberían tenerse en cuenta en el momento de diseñar un protocolo de encaminamiento seguro:

- **Detección de nodos maliciosos:** un protocolo de encaminamiento seguro debería ser capaz de detectar la presencia de nodos maliciosos en la red y debería evitar la participación de dichos nodos en el proceso de encaminamiento. Aún si tales nodos maliciosos participan en el proceso de

descubrimiento de ruta, el protocolo debería elegir caminos que no incluyan a tales nodos.

- **Garantía de descubrimiento de ruta correcta:** si existe una ruta entre los nodos fuente y destino, el protocolo de encaminamiento debería ser capaz de encontrar la ruta y debería asegurar la exactitud de la ruta seleccionada.

- **Confidencialidad de la topología de red:** como ya vimos, un ataque de acceso a la información puede conducir al descubrimiento de la topología de la red por parte de nodos maliciosos. Una vez que la topología de la red es conocida, el atacante puede intentar estudiar el patrón de tráfico en la red. Si algunos de los nodos que son encontrados presentan más actividad que otros, el atacante puede intentar montar, por ejemplo, un ataque por denegación de servicio (DoS) sobre esos nodos que representan cuellos de botella. Esto puede afectar en última instancia al proceso de encaminamiento en curso. Por lo tanto la confidencialidad de la topología de la red es un requerimiento importante a ser cumplido por los protocolos de encaminamiento seguros.

- **Estabilidad contra ataques:** el protocolo de encaminamiento debería ser auto-estable en el sentido de que debería poder volver a su estado de operación normal dentro de límites de tiempo razonables luego de haber sufrido un ataque pasivo o activo. El protocolo debería tener en cuenta que estos ataques no interrumpan permanentemente el proceso y debería asegurar robustez frente a ataques llamados de tipo Binzantino, esto es, que el protocolo debería trabajar apropiadamente aún si algunos de los nodos, los cuales anteriormente habían participado en el proceso de encaminamiento, ahora se convierten en nodos maliciosos o están intencionalmente deteriorados.

En una red ad hoc, desde el punto de vista de un protocolo de encaminamiento, se diferencian dos clases de mensajes: los mensajes de encaminamiento y los mensajes de datos. Ambos tienen distintas necesidades de seguridad. Los mensajes de datos son punto-a-punto y pueden ser protegidos con sistema de seguridad punto-a-punto (como IPSec). Por otro lado, los mensajes de encaminamiento son enviados a vecinos intermedios, procesados, posiblemente modificados, y reenviados. Más aún, como resultado del procesamiento del mensaje de encaminamiento, un nodo puede modificar su tabla de encaminamiento. Esto genera la necesidad de que los nodos intermedios sean capaces de autenticar la información contenida en los mensajes de encaminamiento (necesidad que no existe en las comunicaciones punto-a-punto) para poder aplicar sus políticas de seguridad.

Otra consecuencia de la naturaleza de la transmisión de mensajes de encaminamiento es que, en muchos casos, existirán partes del mensaje que se modifiquen durante su propagación. Esto es muy común en los protocolos por Vector de Distancia, donde los mensajes de encaminamiento usualmente contienen un contador de salto de ruta que ellos solicitan o proveen. Por lo tanto, en un mensaje de encaminamiento se podrían distinguir entre dos tipos de información: información mutable e información no mutable. Se desea que la información mutable en un mensaje de encaminamiento sea segura de tal forma que no necesariamente exista confianza entre los nodos intermedios, aunque, asegurar esta información es mucho más costoso en cómputo.

III. APORTACIÓN DE LA TESIS

A lo largo de este documento hemos definido el concepto de MANET, así como sus características más relevantes para sus posibilidades de implantación comercial. A la vez que hemos expuesto las bondades de su naturaleza autoorganizada y sin necesidad de infraestructuras preexistentes, lo que nos permitiría crear redes en los ambientes más hostiles, también hemos podido comprobar que esta naturaleza trae consigo numerosos problemas. Puesto que la finalidad de cualquier mecanismo creado por el hombre es prestar un cierto servicio o servicios a diversos usuarios, la calidad de dicho mecanismo estará ligada a la calidad del servicio proporcionado por dicho mecanismo. Por ese motivo, el estudio del llamado QoS (calidad de servicio) centra grandes esfuerzos de los investigadores que trabajan en el campo de las redes de comunicaciones. En el caso de las redes ad hoc, obviamente, este interés existe igualmente. Por esto, una vez definido el concepto de red ad hoc, hemos pasado a definir el QoS en redes ad hoc, lo que nos permite clarificar que objetivos debemos cumplir si queremos crear un esquema de redes ad hoc "de calidad". A lo largo de este año, además de profundizar en los conocimientos sobre este tipo de redes y y su situación actual, hemos empezado a trabajar en un esquema de redes ad hoc. En estos momentos, los dos mecanismos que centran nuestros esfuerzos son el protocolo de enrutamiento y el modelo de seguridad. Vamos a pasar a describirlos.

1. Protocolo de enrutamiento

Uno de los principales temas de investigación en los últimos 4 ó 5 años en MANETs ha sido el enrutamiento. El IETF (Internet Engineering Task Force) ha estandarizado varios protocolos de enrutamiento entre los cuales AODV y OLSR son los más reconocidos representantes de los protocolos de encaminamiento reactivos y proactivos. En una MANET, cada nodo se debe comportar como un router, manteniendo individualmente las rutas a otros nodos, por lo tanto, sí el número de nodos crece rápidamente, el número de rutas aumentará también rápidamente impactando tanto en el tamaño de la tabla de encaminamiento como en la búsqueda del camino óptimo. El problema de la escalabilidad (incremento del número de nodos en la MANET) junto con la movilidad de los nodos, puede producir un aumento en la carga (paquetes de control intercambiados por los nodos en la red) generada por el mantenimiento de las rutas, consumiendo el poco ancho de banda disponible en una MANET y por lo tanto reduciendo el desempeño o throughput. Si dejamos de lado la movilidad, el problema es un poco parecido al presentado en los inicios de ARPAnet (precursor del Internet actual), en donde al principio los nodos utilizaban una dirección plana o no jerárquica para su identificación, y por lo tanto cada nodo debía almacenar una ruta para cada uno de los nodos presentes en la red. A medida que ARPAnet seguía creciendo rápidamente, el número de rutas a mantener por cada nodo también aumentaba drásticamente, generando una carga muy grande en la red solamente para el mantenimiento de las rutas.

La solución lógica para reducir esta carga sería utilizar una estructura jerárquica como la definida en Internet, en donde los nodos se agruparan en subredes, de forma que estos nodos aparecen como una sola entrada en la tabla de encaminamiento. Sin embargo, este esquema es difícil de aplicar en MANETs debido a su naturaleza

dinámica y distribuida. El primer problema a solucionar es identificar las modificaciones que deberían plantearse a los protocolos de encaminamiento ad hoc para soportar subredes. Además hay que resolver problemas relacionados con la creación de subredes, asignación de direcciones dinámicas a los nodos, mantenimiento de las sesiones ya establecidas movimiento y detección en el cambio de subred por parte de los nodos. Antes de atacar estos problemas, habría que evaluar si el esfuerzo a realizar vale la pena, es decir, para soportar subredes hay que modificar y proponer cambios en los protocolos, pero, ¿Es factible encontrarse con MANETs que vayan a usar subredes?, ¿Cuál es la ganancia que se obtendrá de este esfuerzo?. Vamos a explicar la propuesta y seguidamente vamos a hacer una evaluación mediante un modelo analítico que justifique nuestra idea.

1.1. Visión general

El objetivo a la hora de diseñar un protocolo de encaminamiento para redes ad hoc, será satisfacer ciertos parámetros de QoS, y que permite que las redes sean escalables. Entre ellos cabe destacar tiempo de servicio, consumo, seguridad de las transmisiones… Nos centraremos en primer lugar en el tiempo de servicio.

A la hora de enviar información entre dos nodos de la red, desearemos que la transmisión cumpla ciertos requisitos:

- Tiempo de transmisión bajo.
- Bajo consumo de recursos.
- Bajo consumo de ancho de banda.

El tiempo de transmisión vendrá determinado por dos factores principales:

- Tiempo que se tarda en descubrir la ruta.
- Tiempo que se tarda en recorrer el trayecto.

Además de estos aspectos, habrá que tener en cuenta el tiempo de respuesta ante errores tras la caída de enlaces.

El tiempo empleado en descubrir la ruta dependerá, principalmente, de si la ruta se conoce previamente, o no. Si la ruta no se conoce, será necesario realizar un proceso de descubrimiento de ruta, con la consecuente inundación de la red de paquetes de control. Para evitar realizar dicho procedimiento, se podría considerar la posibilidad de emplear un enfoque proactivo. Sin embargo, esto limitaría la escalabilidad de nuestra red, puesto que un modelo proactivo es inaplicable en redes grandes por la enorme cantidad de tráfico de control que genera, que en algunos casos puede ser suficiente para saturar la red.

Para evitar esto, nosotros emplearemos un enfoque reactivo para encontrar las rutas por las que deberemos transmitir la información. El principal problema de los mecanismos de descubrimiento de ruta, es que necesitan inundar toda la red para buscar el destino, hasta que, o bien lo encuentran, o bien otro nodo conoce como alcanzarlo. Este proceso de inundación, originará un elevado número de transmisiones de paquetes de control, que podrán provocar colisiones que afectarán aumentarán el tiempo de búsqueda. Sin embargo, si dividimos la red en zonas, cuyos nodos conozcan al resto de los nodos de su zona, podremos mejorar el rendimiento. De esta forma, un nodo S que necesite enviar información a un nodo D, deberá consultar si ese nodo D pertenece a su

zona. En caso contrario, se realizará un proceso de descubrimiento de ruta. Sin embargo, en esta ocasión no será necesario realizar un *broadcast*. Puesto que los nodos conocerán a que zona pertenecen, y que zonas vecinas existen, realizarán una transmisión multicast que permita alcanzar las diferentes fronteras de la zona, evitando generar un excesote tráfico de control, y reduciendo el número de colisiones.

Para conseguir esto, deberemos dividir, en primer lugar, la red en zonas. Para este fin emplearemos una modificación de un algoritmo basado en redes SOM [1] para dibujar grafos en un espacio 2D ϵ [0, 1] x [0, 1]. Cada nodo poseerá la siguiente información.

- Coordenadas propias.
- Coordenadas representativas de la zona.
- Vector desplazamiento respecto al centro de la zona.

B. Enrutamiento intragrupal.

El objetivo de nuestro algoritmo será estar en situación de poder proporcionar una respuesta rápida sobre la pertenencia o no pertenencia de una entidad a la subred.

El caso óptimo sería que cada nodo mantuviera una tabla con todos los nodos pertenecientes a la red, que se mantuviera actualizada constantemente. Esto presentaría dos problemas principales:

- La necesidad de enviar paquetes de control periódicamente (comportamiento proactivo). Esto conllevará una sobrecarga de tráfico en la red.

- Puesto que el criterio para constituir una subred es el diámetro, en el caso de que la densidad de nodos sea muy alta, el número de entidades de la red será muy alto. Por lo tanto, no se podrán almacenar toda la información de la red, puesto que el gasto de memoria sería excesivo.

Para tener un acceso rápido a la lista de nodos que componen la subred, que se encontrará distribuida en diferentes entidades, utilizaremos aproximaciones de algoritmos de recorrido de grafos. Consideramos el algoritmo de Floyd, que devuelve los caminos más cortos entre cada par de nodos, considerando que el peso de cada arista (conexión) es 1, por lo que la métrica será equivalente al número de saltos. Cuando deseemos consultar desde un nodo, la pertenencia o no de otra entidad a la red, enviaremos un *Membership Request Packet,* que alcanzará a todos los nodos accesibles a un salto. Posteriormente, sólo será retransmitido por un número de nodos suficiente para que el paquete llegue a todos los nodos a 2 saltos, y así sucesivamente. Se elegirán nodos que, además, nos proporcionen garantías de cubrir la red en un número reducido de saltos. Para ellos se utilizará una función heurística, calculada a partir de la media y la varianza de las distancias hayadas con el algoritmo de Floyd. De esta forma, se premia a los nodos cuya distancia media al resto de entidades del grupo sea pequeña, teniendo en cuenta que esta distancia sea uniforme, de forma que se recorra toda la red con el mínimo número de retransmisiones y de saltos. Además, debido a que los vecinos que se encuentren a una distancia menor que r_{min} (valor configurable), conocerán su existencia mediante el empleo de mensajes *Hello*, se podrá acceder a la información compartida de forma muy eficiente, sin necesidad de mantenerla íntegra en todos los equipos.

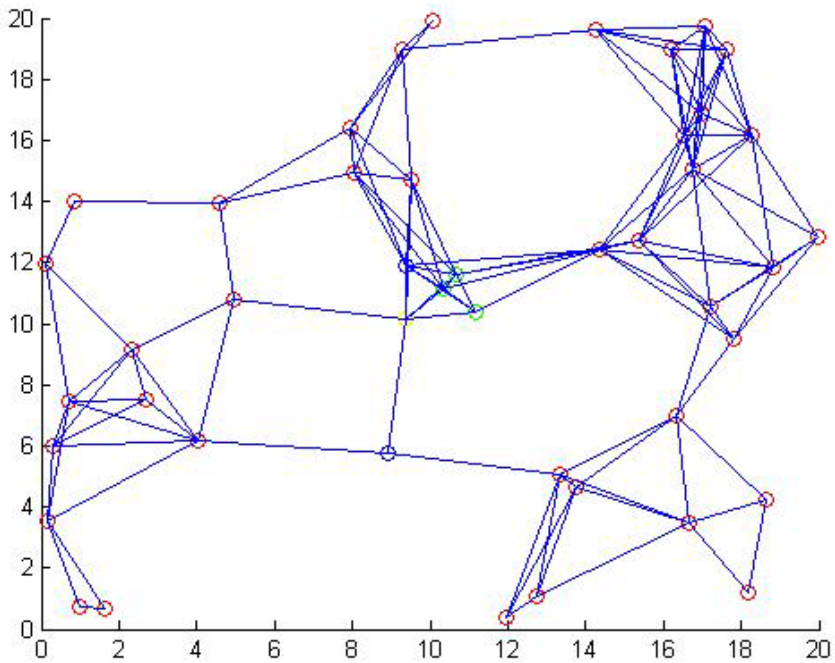
El mayor problema para aplicar esta solución, será implementar el algoritmo de Floyd, cuya complejidad y sobrecarga serían excesivas. Por ello, nos limitaremos a tomar medidas estimadas a partir de los paquetes de control y de datos que circulan en la red, para asignar un peso a cada enlace. Posteriormente, los nodos se comportarán de forma similar a las redes neuronales competitivas.

A continuación mostramos un ejemplo de la aplicación del protocolo a una subred de 40 nodos distribuidos de forma aleatoria en una región bidimensional R = $[0, 20]^2$. El rango es variable, por lo que existirán enlaces unidireccionales. Se distinguen 4 grupos de nodos según los resultados obtenidos por la función heurística:

- Amarillo: el nodo con menor valor.
- Verde: grupo de nodos con valores cercanos al mejor.
- Azul: grupo de nodos con valor mayor que los vedes, siendo la diferencia menor a 0,2.
- Rojos: el resto de nodos que forman la subred.

Los tres primeros grupos de nodos, serán los usados para diseminar.

Si consideramos que r_{min} vale 2, para conocer la presencia de un nodo a la subred desde, por ejemplo, el nodo situado en las coordenadas (18,1), en la parte inferior derecha, se necesitará en el peor de los casos, 5 saltos, siendo la media cercana a 3. Y lo más importante, se cubriría la red realizando solamente cinco transmisiones en



el proceso de *Request*, más la devolución unicast del mensaje de pertenencia. Si no se recibe nada tras un tiempo que será proporcional al diámetro estimado del grupo, se considerará que el nodo no pertenece a la red.

B. Enrutamiento intergrupal.

Las subredes básicas (las formadas por nodos) se agruparán en subredes de nivel 2. Generalizando, las subredes de nivel i, se agruparán en subredes de nivel i + 1. La red

completa será la subred de mayor índice. Con esto tenemos, que el tráfico entre grupos diferentes se tratará como el tráfico intragrupal. Sin embargo, al existir distancias mayores entre nodos, y un número más elevado, necesitaremos modificar las heurísticas empleadas con el fin de obtener una mayor optimización.

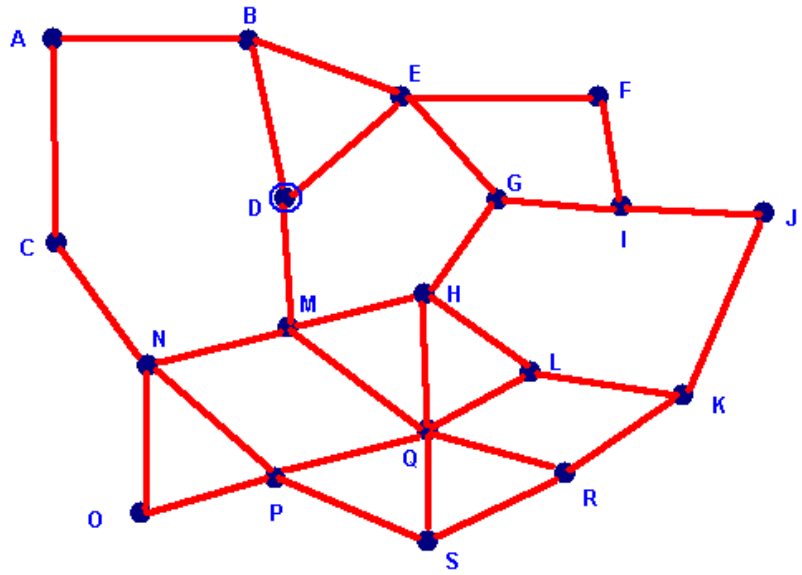
Para alcanzar desde el nodo de una subred el exterior de la subred, se realizará un envío multicast a los nodos *confluencia*. A priori, enviar un mensaje multicast de manera eficiente sería un tema difícil. Sin embargo, en este caso los nodos de la subred conocen los destinos del mensaje, por lo que no es necesario incluir en el mensaje sus direcciones, ni realizar un proceso de establecimiento de ruta. De esta forma, el envío multicast se hará de manera muy eficiente.

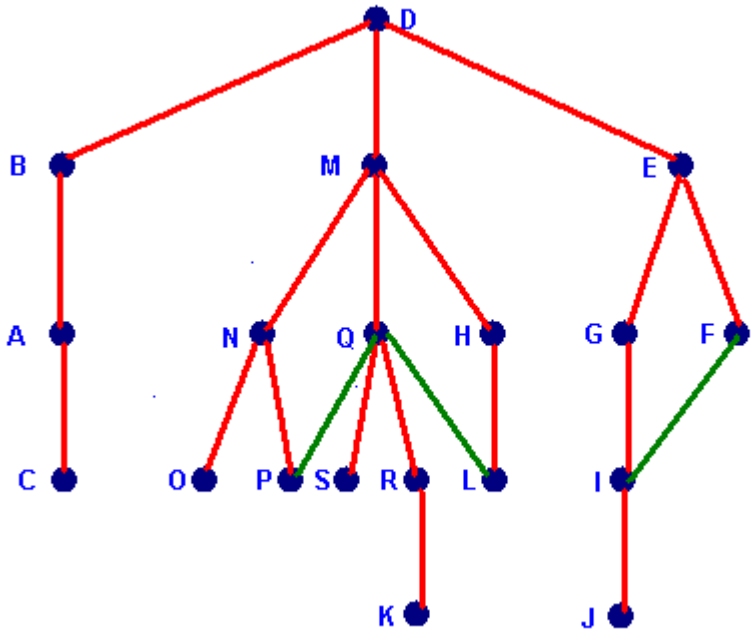
1.1.1. Broadcast

En una red autoorganizada que no tiene ninguna infraestructura externa que le de algún soporte necesita tener mecanismos que proporcionen ciertos servicios como configuración de direcciones, administración de claves, etc. En estos mecanismos nombreados, y otros muchos, vemos que se necesita mantener la unicidad de un tipo de datos y cumplir ciertas normas que involucran a toda la red. Cuando tenemos servidores DHCP en redes cableadas, o tenemos una autoridad certificadora central, este problema no existe, puesto que la autoridad certificadora sabe que claves ha emitido ya. Ocurre lo mismo con el servidor DHCP que sabe que direcciones ha asignado ya dentro de su rango. En este caso, como no tenemos unos servidores centrales, la red debe cooperar para mantener la unicidad de estos datos. Por lo tanto, se hace necesario poder realizar consultas eficientemente. La manera más rápida de hacer un consulta en toda la red sobre la existencia o no de una dirección es la inundación de la red mediante un envío broadcast. Sin embargo, eso hace que todos los nodos tengan que retransmitir mensajes aunque no contengan información importante. Además, los nodos cercanos entre sí no aportarán mucha información nueva, y si aportarán mucha información redundante, sobre todo en esquemsa proactivos.

¿Qué solución podemos encontrar para este problema?. Nuestro protocolo de enrutamiento jerárquico creaba subredes. Y estas subredes tienen una cohesión importante, con lo que se puede conocer la información completa de la red de una forma rápida. Es decir, al crear subredes estamos creando centros de información. Por lo tanto, primera ventaja, nestro broadcast no funciona a nivel de nodos, sino que trabaja a nivel de subredes. Una vez alcanzada la subred, la subred responde como un ente único. Ya sea contestando una solicitud de información, o bien reenviando a las redes vecinas para continuar el proceso de broadcast.

Pero adicionalmente, este comportamiento se puede mejorar, tanto a nivel de nodos como a nivel de subredes.





Como vemos, cuando D intenta hacer un broadcast, se forma un árbol de envío. Este árbol parte del nodo que emite el broadcast y su objetivo es inundar toda la red hasta llegar a los periféricos. En el árbol se muestra un comportamiento ideal en el que los nodos avanzan hacia el exterior. Pero realmente se producen muchas retransmisiones que necesitan ser controladas en zonas ya cubiertas, y además, como la señal se propaga en círculo se producen numerosas colisiones en nodos a la misma distancia del origen. Este problema se reduce con la técnica de los Multi Point Relays. Sin embargo, esta mejora depende de la buena elección de los reales. Para mejorar nuestra capacidad de decisión en este aspecto, así como para mejorar la selección de las subredes y la previsión de los cambios topológicos, estamos trabajando con datos geométricos y

geográficos virtuales, calculados mediante técnicas adaptadas de la redes neuronales y la lógica fuzzy. Exponemos alguno de los conceptos que estamos estudiando:

- Coordenadas virtuales.
- Ángulo de visión de un nodo.
- Ángulo de cobertura de un enlace.
- Grado de excentricidad.
- Profundidad del ángulo virtual.

2. Seguridad

2.1. Autoridad de certificación

En las redes ad hoc no tenemos ninguna autoridad de certificación central que emita certificados. De ser así, la obligación de la existencia de una autoridad central limitaría la libertad de implantación de una MANET, que es su principal ventaja. Por lo tanto, al igual que toda la resposabilidad del enrutamiento recae sobre los propios terminales, estos tendrán que hacer también las veces de autoridad de certificación. Vamos a reflexionar sobre como los nodos participantes en la red pueden ejercer como autoridad de certificación.

En primer lugar habría que desechar el concepto de autoridad de certificación central. Si continuamos con este modelo, sería necesario escoger un nodo o un grupo de nodos que ejercieran constantemente como autoridad certificadora central. Esto obligaría a que estos nodos permanecieran siempre conectados, puesto que si dejaran de pertenecer a la red habría que buscarles un sustituto. Además, recibirían un número de consultas demasiado elevado, con el consecuente gasto de recursos. Y por último, esto obligaría o bien, a conocer una ruta a dichas autoridades con el consecuente gasto de memoria, o bien cada nodo que se uniera a la red debería realizar un descubrimiento de ruta para localizar la autoridad, con el consiguiente retardo.

Por todos estos inconvenientes que hemos mencionado, definimos el concepto de autoridad on-line. En el esquema de autoridad on-line cada nodo almacena su par del claves pública y privada. La autoridad controla la pertenencia de los nodos a la red y decide quénodos pueden unirse a la red. En un escenario con autoridad on-line, el par de calves de un nodo puede ser generado por el mismo nodo o por la autoridad, pero en ambos casos la autoridad debe emitir un certificado que legitima al nodo para participar en la red. La clave pública que obtenga el nodo estará ligada a su dirección, no pudiendo existir otro nodo en la red con la misma clave pública.

Ahora bien, ¿Quién emite el certificado y cómo comprobamos que el certificado presentado por un nodo es válido?. Vamos a tomar una idea del popular Pretty Good Privacy (PGP). En PGP cada usuario distribuye certificados a sus conocidos y confía en sus amigos como presentadores. Es decir, si un nodo A certifica a un nodo B, y B ha certificado a un nodo C, A confiará en C. De esta forma, nosotros proponemos la creación de un grafo de certificados para certificar en MANETs. Definamos las características de este modelo de certificación:

- Los nodos emiten y distribuyen sus propios certificados y firman certificados de los demás.
- El modelo supone la existencia de confianza previa entre algunos nodos y genera confianza nueva entre los nodos de forma PGP.
- Cada nodo A tiene una clave pública, una clave privada, un repostorio local de certificados de todos los nodos en los que se confía y una lista de certificados de todos los nodos que confían en ella. Así, cada certificado es almacenado dos veces, por su emisor y por su dueño.
- Antes de que un certificado caduque, su emisor debería distribuir una versión actualizada con fecha de caducidad extendida.
- La autenticación de claves se realiza mediante una cadena de certificados: cuando u quiere obtener la clave pública de v, busca una cadena de certificados para los nodos entre ambos de forma que:
    - El primer certificado de la cadena puede ser comprobado directamente por u.
    - Cada uno de los demás certificados puede comprobarse usando la clave pública contenida en el certificado previo de la cadena.
    - El último certificado contiene la clave pública de usuario destino v.
    - U comprueba que todos los certificados de la cadena son:
        - Válidos (no revocados)
        - Correctos (no falsos)

Para encontrar cadenas de certificados apropiadas, cada nodo mantiene dos repositorios locales de certificados: los no actualizados y los actualizados. Los no actualizados dan una buena estimación sobre el grafo certificado y la elección de los certificados en el repositorio actualizado se realiza con un algoritmo adecuado.

Este modelo presenta un problema fundamental: un nodo aislado es el resposable de autorizar o desautorizar a los nodos que se unen a la red en su entorno. Esto puede hacer que un nodo se haga pasar por un nodo bienintencionado comportándose correctamente, pero que emita certificados incorrectos a otros nodos maliciosos, de forma que estos nodos puedan suplantar la identidad de miembros de la red. Además, el hecho de que el emisor derl certificado sea necesario para completar la autenticación, hace que si se cae dicho emisor, no podamos autenticar al nodo, que deberá solicitar otra vez un certificado. Además, esta forma de actuar es un poco contradictoria con el espíritu cooperativo de estas redes.

Por lo tanto, vamos a mejorar este modelo. Para ello, obligaremos a firmar a un número n de componentes de la red con una firma conjunta. Esa firma se repartirá entre varios componentes de la red. Lo ideal sería que no bastara con un solo terminal para poder certificar al nuevo nodo, sino que varios tuvieran que certificar al nodo. Pero también deberíamos poder seguir certificando al nodo aunque alguno de los nodos certificantes saliera de la red. ¿Es esto posible?. Sí, es posible usando un esquema de compartición de secretos. La compartición de secretos se basa en la existencia de problemas matemáticos en los que es muy fácil calcular el resultado teniendo al menos n muestras, pero es imposible de resolver teniendo menos de n muestras. La idea en nuestro modelo es generar un numero t mayor de n pero y distribuirlos entre diferentes subredes, cumpliendo la regla de que el número de muestras asignado a una subred i, m_i es menor que 2*(n /m), siendo m el número de subredes implicadas.

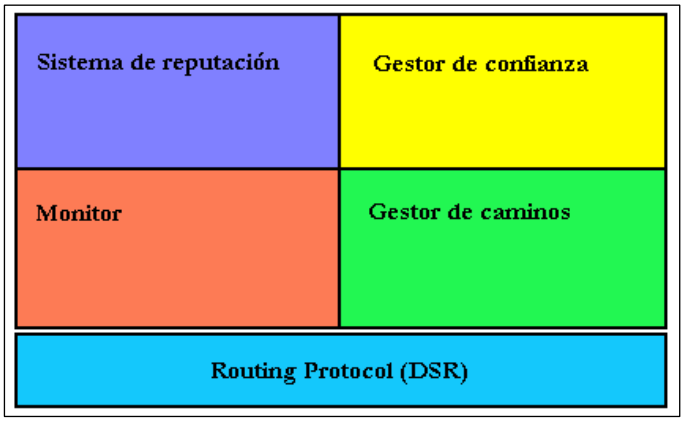
2.2. Detección de intrusos. Modelo cooperativo.

Características:

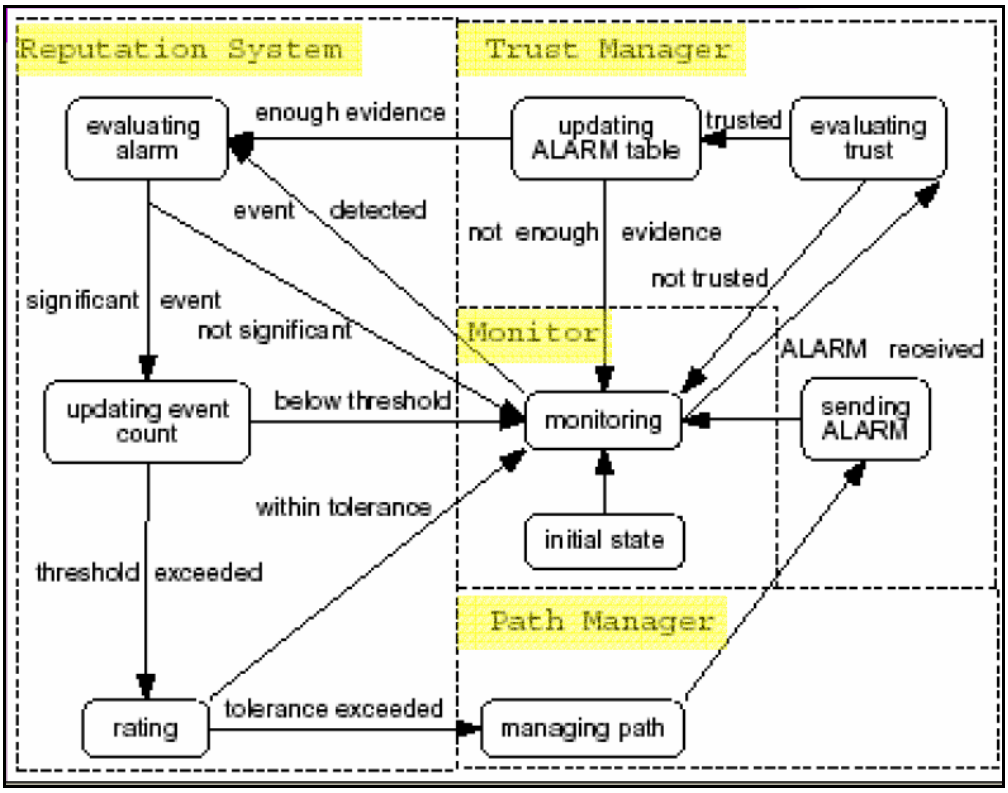
- Trata de detector nodos atacantes mediante la combinación de monitorizado y el establecimiento de rutas que eviten nodos egoístas
- Causa una reacción en los demás nodos que se traduce en una desventaja para el nodo egoísta ya que los paquetes de los nodos atacantes no son enviados por los nodos honestos.
- Las relaciones de confianza y las decisiones sobre el enrutamiento se basan en las experiencias propias, observaciones de otras experiencias, e informes sobre comportamientos de otros nodos.
- Soporta hasta un 60 % de nodos egoístas,
- Si un nodo es acusado erróneamente, o se arrepiente y se comporta de forma honesta durante cierto tiempo, se lleva a cabo su re-socialización y reintegración en la red.

Componentes presentes en cada nodo:

- Sistema de reputación.
- Monitor.
- Gestor de confianza.
- Gestor de caminos.



Se basa en DSR.



El monitor es el vigilante del vecindario:

- Los nodos con más probabilidad de detectar un comportamiento incorrecto son los nodos del vecindario del atacante, y también la fuente y/o el destino si detectan un comportamiento inusual o no obtienen respuestas adecuadas.

- Los nodos pueden detectar desviaciones mediante experiencias propias o la escucha de la transmisión del siguiente nodo en la ruta, ya que si se mantiene una copia de un paquete mientras se escucha la transmisión del siguiente nodo, cualquier cambio de contenido puede ser detectado.

- Registra las desviaciones del comportamiento normal, y en cuanto ocurre un mal comportamiento, llama al gestor de confianza y al sistema de reputación.

El gestor de confianza:

- Trabaja con mensajes ALARMA entrantes y salientes.

- Los mensajes ALARMA son enviados por el gestor de confianza de un nodo para avisar a otros sobre la presencia de nodos atacantes, tras haber experimentado, observado o recibido un informe de comportamiento deshonesto.

- Los receptores de estos mensajes se llaman amigosm y son administrados en una lista de amigos.

- Cada nodo siempre debe comprobar el nivel de confianza de la fuente de una ALARMA antes de producir una reacción.

- Para la gestión de confianza de esas ALARMAs se usa un mecanismo similar al gestor de confianza definido en PGP.

- Se examina el nivel de confianza de todas las firmas adjuntas a un mensaje, y se calcula una puntuación de la validez, P.e., dos firmas marginalmente confiables pueden ser equivalente a una sola firma de total confianza. El esquema es ajustable de forma que puede requerir un número diferente de firmas marginalmente confiables para igualarla a una clave válida.

El gestor de confianza consiste en los siguientes componentes:

- Una tabla de alarmas con información sobre alarmas recibidas.

- Una tabla de confianza que gestiona los niveles de confianza de cada nodo para determinar la confianza de una alarma.

- Una lista de amigos con todos los amigos a los que un nodo potencialmente envía alarmas.

Una vez comprobado el nivel de confianza de una alarma, se envía al sistema de reputación.

El sistema de reputación gestiona una tabla de nodos y sus puntuaciones:

- Hace uns tasación de la calidad de los nodos.

- Para evitar una tasación centralizada, se mantienen listas de tasaciñon locales y/o listas negras en cada nodo, que se intercambian con amigos.

- En las rutas enviadas, el nodo origen puede incluir ovejas negras a evitar en el enrutamiento, que también alarma a los nodos intermedios.

- Los nodos comprueban los nodos con mala puntuación en la lista negras antes de enviar algo. El problema de distinguir entre nodos atacantes acusados y demostrados, e.d., evitar falsas acusaciones, puede disminuirse mediante listas de recuperación de nodos que se han portado bien durante un período especificado de tiempo.

- Se supone que el mal comportamiento será la excepción, por lo que el sistema de reputación se construye por experiencia negativa en lugar de por impresiones positivas.

- La puntuación de un nodo baja sólo si hay suficiente evidencia de mal comportamiento, y cuando ha ocurrido un número de veces que excede un umbral. La tasación es entonces cambiada de acuerdo con una función que asigna diferentes pesos al tipo de detección de comportamiento, p.e. mayor peso a la propia experiencia, y menor peso a las observaciones del vecindario, y menor aún a la experiencia informada.

- Si la tasación de un nodo se ha deteriorado tanto que ha caído por debajo de un rango tolerable, se activa el gestor de caminos.

El gestor de caminos tiene las siguientes funciones:

- Ordenación de caminos según la métrica escogida p.e., reputación de los nodos del camino.

- Borrado de caminos con nodos atacantes.

- Accióm tras la recepción de una petición de ruta de un nodo atacante (p.e. ignorar, no enviar ninguna respuesta).

- Acción tras la recepción de petición de ruta hacia un nodo atacante (p.e. ignorar, alertar a la fuente).

Las ideas expuestas aquí está aún en período de desarrollo, pero de momento presentan resultados priometedores. Hasta ahora, estos trabajos se han visto culminados en la presentación de esta tesis, asi como los artículos: “Optimización de Redes Ad Hoc mediante el algoritmo Ant Colony Optimization” y “Optimización Del Tráfico En Redes Ad Hoc Mediante Protocolos Jerárquicos”.

APÉNDICE A

Teoría de números

Lema 1. Si a, b, c y r son números enteros no nulos tales que a = cb + r, se tiene que mcd(a,b) = mcd(b,r).

Demostración. Denotamos mcd(a,b) por p y mcd(b,r) por q. Como p|a y p|b, se tiene que a = tp y b = sp. Por tanto, r = a – bc = tp – csp = (t – cs)p; esto es, p también divide a r, lo que implica que $p \leq q = mcd(b, r)$.

Pero, a su vez, b = uq, r = sq, puesto que q|b y q|r; entonces a = cb + r = (cu + s)q. Por tanto q|a, por lo que $q \leq p = mcd(a,b)$. Así pues, p = q.

Lema 2. Si a y b son dos números enteros no nulos tales que b|a, entonces mcd(a,b) = |b|.
Demostración. Es inmediata, pues b es un divisor común de a y b y cualquier otro divisor común de a y b verifica n < |b|.

Algoritmo de Euclides. Sean a $\in$ Z, b $\in$ Z, a $\neq$ 0, b $\neq$ 0; el algoritmo de la división implica la existencia de dos números naturales c_0 y r_1 tales que

$a = c_0b + r_1, 0 \leq r_1 < |b|$.

Si $r_1 = 0$, b|a y mcd(a,b) = |b|, por el lema anterior; si, por el contrario, $r_1 \neq 0$, el lema nos permite deducir $mcd(a, b) = mcd(b,r_1)$, y aplicamos de nuevo el algoritmo de la división encontrando dos números naturales c_1 y r_2 tales que

$b = c_1r_1 + r_2, 0 \leq r_2 < |r_1|$.

Este proceso puede continuarse de manera que en el paso i obtengamos

$r_{i-2} = c_{i-1}r_{i-1} + r_i, 0 \leq r_i < |r_{i-1}|$

Puesto que $|b| > |r_1| > |r_2| \ldots \geq 0$, es evidente que en un número finito n de pasos, con $n \leq |b|$, llegaremos a un resto $r_n = 0$, es decir $r_{n-2} = c_{n-1}r_{n-1}$. Ello implica que $r_{n-1}|r_{n-2}$ y por el lema anterior $mcd(r_{n-2}, r_{n-1}) = |r_{n-1}|$. Como

$mcd(a,b) = mcd(b,r_1) = \ldots = mcd(r_{n-2}, r_{n-1}) = |r_{n-1}|$.

Corolario 1. Propiedad lineal del máximo común divisor: Si a y b son dos números enteros no nulos y p = mcd(a,b), existen dos números enteros u y v tales que p = ua + vb.

Demostración. Supongamos que $p = r_{n-1}$. Puesto que para todo i, $r_{i-2} = c_{i-1}r_{i-1} - r_i$, de acuerdo con las fórmulas obtenidas en la descripción del algoritmo de Euclides, se obtiene

$r_{n-1} = r_{n-3} - c_{n-2}r_{n-2} = u_{n-1}r_{n-3} - v_{n-1}r_{n-2} = u_{n-1}r_{n-3} + v_{n-1}(r_{n-4} - c_{n-3}r_{n-3}) = u_{n-2}r_{n-4} + v_{n-2}r_{n-3} = \ldots = u_i r_{i-2} + v_i r_{i-1} = \ldots = u_1 a + v_1 b,$

donde los u_i, v_i son enteros.

Corolario 2. Dos números enteros no nulos a y b son relativamente primos si y sólo si existen dos números enteros u y v tales que ua + vb = 1.

Logaritmos

Problema del logaritmo discreto. Sea G un grupo finito, por ejemplo, $G = (Z/p)^x$. Dado b Є G y una potencia a de b, se desea encontrar el entero positivo más pequeño talque $b^n = a$. Por lo tanto, el problema del logaritmo discreto es el problema de calcular $n = \log_b(a)$ para a,b Є G.

BIBLIOGRAFÍA Y REFERENCIAS

[1] Gómez C., Paradells J., "Redes ad-hoc: el próximo reto", Buran nº 21, págs. 30 a 37. Mayo 2004.

[2] Jonson D., Maltz D., Hu Y., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)" IETF Internet-Draft draft-ietf-manet-dsr-10.txt. Julio 2004.

[3] Chakeres I., Perkins C., "Dynamic MANET On-demand (DYMO) Routing" IETF Internet-Draft draft-ietf-manet-dymo-01.txt. Marzo 2006.

[4] Macker J., "Simplified Multicast Forwarding for MANET", IETF Internet-Draft draft-ietf-manet-smf-01.txt. Marzo 2006.

[5] Kim K., Montenegro G., Daniel Park S., Chakeres I., Yoo S., "Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing", IETF Internet-Draft draft-montenegro-6lowpan-dymo-low-routing-00.txt. Octubre 2005.

[6] Kim K., Montenegro G., Daniel Park S., Yoo S., Kushalnagar N., "6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)", IETF Internet-Draft draft-daniel-6lowpan-load-adhoc-routing-02.txt. Marzo 2006.

[7] TinyAODV Implementation, TinyOS Source Code Repository http://cvs.sourceforge.net/viewcvs.py/tinyos/tinyos-1.x/contrib/hsn/.

[8] Perkins C., Belding-Royer E., Chakeres I., "Ad Hoc On-Demand (AODV) Routing, IETF Internet-Draft draft-perkins-manet-aodvbis-01.txt. Enero 2004.

[9] Chakeres, I., Klein-Berndt L., "AODVjr, AODV Simplified", ACM SIGMOBILE Mobile Computing and Communications Review pp. 100- 101, Julio 2002.
Web NST-AODV, URL
http://www.i2cat.net/i2cat/servlet/I2CAT.MainServlet?seccio=6_5_2

[10] C. Gomez, P. Salvatella, O. Alonso, J. Paradells, 'Adapting AODV for IEEE 802.15.4 Mesh Sensor Networks: Theoretical Discussion and Performance Evaluation in a Real Environment', 7th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM 2006), Niagara Falls, USA. Junio 2006.

[11] Akyildiz I., Wang X., Wang W., "Wireless Mesh Networks: a survey", Computer Networks Journal (Elsevier), 47(4), 2005.

[12] Chen B., Muniswamy-Reddy K., and Welsh M., "Lessons Learned from Implementing Ad-Hoc Multicast Routing in Sensor Networks", Harvard University Technical Report TR-22-05, Noviembre 2005.

[13] Draves R., Padhye J., Zill B., "Comparison of routing metrics for static multi-hop wireless networks", ACM SIGCOMM Computer Communication Review, v.34 n.4, Octubre 2004.

[14] Adya A., Bahl P., Padhye J., Wolman A., and Zhou L., "A multi-radio uni.cation protocol for IEEE 802.11 wireless networks". In BroadNets, 2004.

[15] De Couto D., Aguayo D., Bicket J., and Morris R., "High-throughput path metric for multi-hop wireless routing". In *MOBICOM*, Septiembre 2003.

[16] Hu Y. and Jonson D. B., "Design and demonstration of live audio and video over multi-hop wireless networks". In MILCOM, 2002.

[17] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad hoc On-Demand Distance Vector (AODV) routing. Internet Request for Comments RFC 3561, Nov. 2003. http://moment.cs.ucsb.edu/pub/rfc3561.txt

[18] M. Guerrero Zapata. Secure Ad Hoc On-Demand Distance Vector (saodv) routing, Sep. 2006. INTERNET-DRAFT draft-guerrero-manet-saodv-06.txt, http://tools.ietf.org/id/draft-guerrero-manet-saodv-06.txt

[19] C. Siva Ram Murthy, B.S. Manoj. AdHoc Wireless Networks. Archiectures and Protocols. Prentice Hall, 2004. ISBN 0-13-147023-X.
Mohammad Ilyas. The Handbook of Ad Hoc Wireless Networks. CRC Press, 2003. ISBN 0-8493-1332-5.

[20] M. Guerrero Zapata. Secure Ad hoc On-Demand Distance Vector Routing. ACM Mobile Computing and Communications Review (MC2R), 6(3):106–107, July 2002.
http://portal.acm.org/ft_gateway.cfm?id=581312&type=pdf

[21] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifiable (SUCV) identifiers and addresses. Network and Distributed System Security Symposium (NDSS '02), Feb. 2002.
http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/monten.pdf

[22] http://www.wikipedia.org

[23] M. Guerrero Zapata and N. Asokan. Securing Ad hoc Routing Protocols. In Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002), pages 1–10, September 2002.
http://www.cs.huji.ac.il/labs/danss/sensor/adhoc/routing/zapta_2002securingadhocrouting.pdf

[24] Huaizhi Li and Mukesh Singhal. A Secure Routing Protocol for Wireless Ad Hoc Networks. In Proceedings of the 39th Hawaii International Conference on System Sciences, pages 225.1, January 2006.
http://doi.ieeecomputersociety.org/10.1109/HICSS.2006.29

[25] M. Guerrero Zapata. Securing and Enhancing Routing Protocols for Mobile Ad hoc Networks. Doctoral Thesis. Computer Architecture Department. University of Catalonia, 2006.
http://www.tdx.cbuc.es/TDX-1212106-103102/

[26] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. SCS Communication Networks and Distributed Systems Model-ing and Simulation Conference (CNDS 2002), Jan 2002.
http://citeseer.ist.psu.edu/rd/9909163%2C559572%2C1%2C0.25%2CDownload/http://coblitz.codeen.org:3125/citeseer.ist.psu.edu/cache/papers/cs/27088/http:zSzzSzwnl.ece.cornell.eduzSzPublicationszSzmc2r_10_02.pdf/papadimitratos02secure.pdf

[27] Y. Lin, A. Hamed Mohsenian Rad, Vincent W. S. Wong, Joo-Han Song and Joo-Han song. Experimental comparisons between SAODV and AODV routing protocols. In International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems and Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling, pages 113-122, October 2005.
http://portal.acm.org/citation.cfm?id=1089757

[28] M. Guerrero Zapata. Simple ad hoc key management (sakm), Feb. 2006. INTERNET-DRAFT draft-guerrero-manet-sakm-00.txt
http://tools.ietf.org/id/draft-guerrero-manet-sakm-00.txt

[29] C. Madson and R. Glenn. The use of HMAC-MD5-96 within ESP and AH. Internet Request for Comments RFC 2403, Nov. 1998.
http://rfc.net/rfc2403.html

[30] C. Madson and R. Glenn. The use of HMAC-SHA-1-96 within ESP and AH. Internet Request for Comments RFC 2404, Nov. 1998.
http://rfc.net/rfc2404.html

[31] Y. Zhang, W. Lee. "Intrusion detection in wireless adhoc networks". Mobile Computing and Networking,
275-283, 2000.

[32] O. Kachirski, R. Guha. “Intrusion detection using mobile agents in wireless ad hoc networks“. Proceedings of the
IEEE Workshop on Knowledge Media Networking, 153 -158, 2002.

[33] J. Lundberg. "Routing security in ad hoc networks".

[34] V. Kärpijoki. "Security in ad hoc networks".

[35] R. Ramanujan, A. Ahamad; J. Bonney, R. Hagelstrom, K. Thurber. “Techniques for intrusion-resistant ad hoc
routing algorithms (TIARA)”. Proceedings of IEEE Military Communications Conference (MILCOM'00), vol.2, Los Angeles, CA, USA, 22- 25, 2000.

[36] R. Ramanujan, S. Kudige, S. Takkella, T. Nguyen, F. Adelstein. "Intrusion-resistant ad hoc wireless networks". MILCOM 2002. Proceedings , vol 2 , 890 -894, 2002.

[37] P. Papadimitratos, Z. J. Haas. "Secure routing for mobile ad hoc networks". Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation (CNDS), 2002.

[38] Y-C. Hu, D. B. Johnson, A. Perrig. "Secure efficient distance vector routing in mobile wireless ad hoc networks".
Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 2002.

[39] Y-C. Hu, A. Perrig, D. B. Johnson. "Ariadne: a secure ondemand routing protocol for ad hoc networks". Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (MobiCom), 2002.

[40] L. Zhou, Z. J. Haas. "Securing ad hoc networks". IEEE Networks, vol. 13, issue 6, 1999.

[41] H. Luo, S. Lu. "Ubiquitous and robust authentication services for ad hoc wireless networks". Technical Report 200030, UCLA Computer Science Department, 2000.

[42] J-P. Hubaux, L. Buttyán, S. Capkun. "The quest for security in mobile ad hoc networks". ACM, 2001.

[43] D. Balfanz, D. K. Smetters, P. Stewart, H. Chi Wong. "Talking to strangers: authentication in ad-hoc wireless
networks". Internet Society, Conference Proceeding of NDSS Conference 2002.

[44] N. Asokan, P. Ginzboorg. "Key agreement in ad hoc networks". Computer Communications, vol. 23, 2000.

[45] Jiangchuan Liu, Kazem Sohraby, Qian Zhang, Bo Li, and Wenwu Zhu, "Resource Discovery in Mobile Ad Hoc Networks"

[46] Stefano Basagni, Marco Conti, Silvia Giordano, Ivan Stojmenovic, Mobile Ad Hoc Networking

[47] Ramin Hekmat, Ad-Hoc Networks: Fundamental Properties and Network Topologies

[48] Paolo Santi, Topology Control in Wireless Ad Hoc and Sensor Networks

[49] Amitabh Mishra, Security and Quality of Service in Ad Hoc Wireless Networks

[50] Shin-Lin-Wu, Yu-Chee Tseng, Wireless Ad Hoc Networking